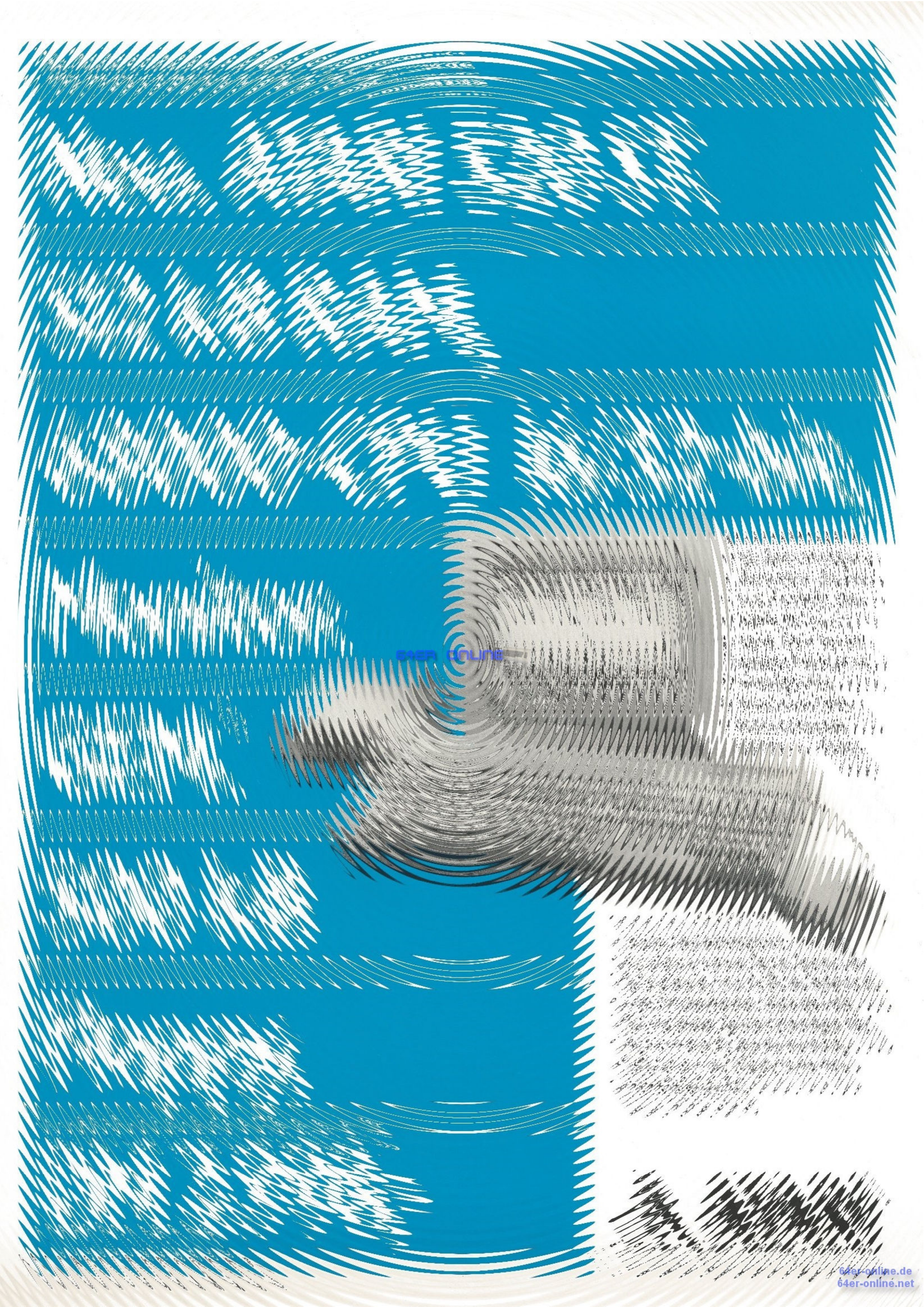
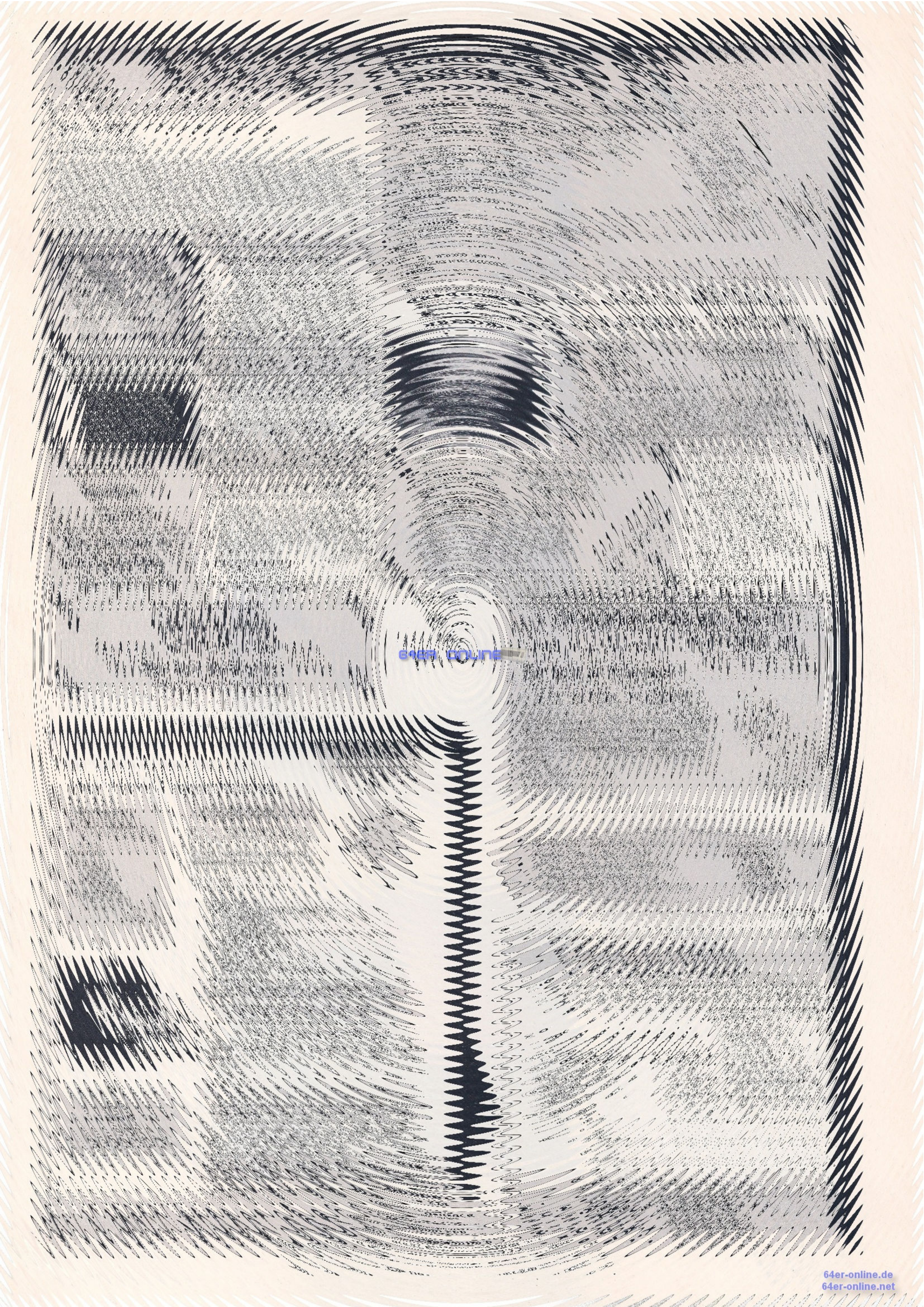


SEPTEMBER 1985
ÖS 55,—/sfr. 6,50
Lit 5.500/hfl 8,-/dkr 30,-
DM 6,50
64'er
9|85
DAS MAGAZIN FÜR COMPUTER-FANS
Eine Markt & Technik Publikation
Musik und C 64
★ Grundlagen für Anfänger
★ Vergleichstest Musikprogramme
Test: Spitzen-Drucker unter 1000 Mark
Software-Test
★ C und Forth: Sprachen für Profis
★ Der Preisbrecher: Textprogramm für 64 Mark
★ Vergleich: Grafik-Erweiterungen
Tips & Tricks für C 64, C 16 und VC 20
Software für 10000 Mark zu gewinnen
Neu: Zum Sammeln 64'er-Extra
Die unentbehrliche Hilfe für Programmierer
6581 2283
diminuendo
36
64er-online.de
64er-online.net





Seite 162



Das neue Urheberrecht-Gesetz

Das Urheberrecht-Gesetz findet nun auch Anwendung auf Computer-Programme. Was darf man kopieren und was nicht? Welche Strafen riskiert man beim »Klauen« von Software? Nur zwei Fragen, die wir Ihnen in einem Artikel zum neuen Urheberrecht-Gesetz beantworten.
Seite 162

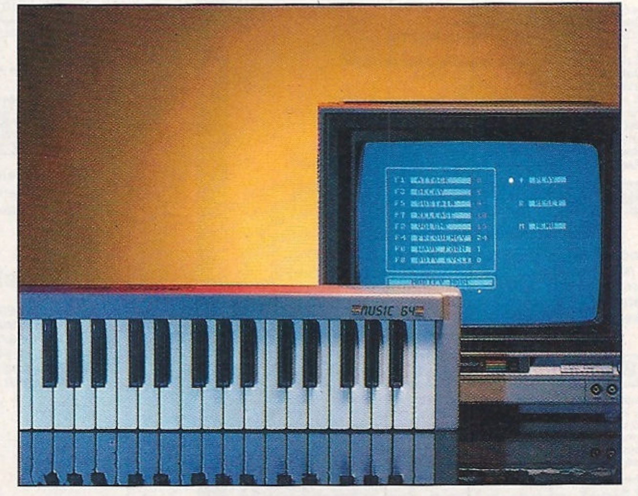
Seite 132



Test: drei neue Drucker

Der stetige Trend zu besserer Schriftqualität bei Matrixdruckern ist unverkennbar. Gleichzeitig bieten immer mehr Firmen Drucker an, die sich direkt an den C 64 anschließen lassen. Der Panasonic KX-P1091, der Riteman C+ und der Star SG 10C sind Vertreter dieser neuen Druckergeneration.
Seite 132

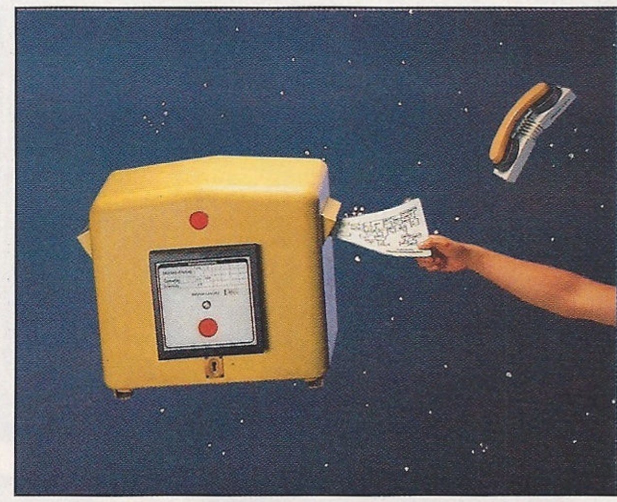
Seite 17



Musik mit dem C 64

Im Musikschwerpunkt befassen wir uns mit dem Angebot an »klingender« Hard- und Software. Sie erfahren, was es da für den C 64 gibt und für wen welches Zubehör interessant ist. In einem Grundlagen-Artikel alles über die SID-Programmierung, einschließlich einer Reihe von Demo-Soundeffekten und natürlich Listings zum Abtippen.
Seite 17

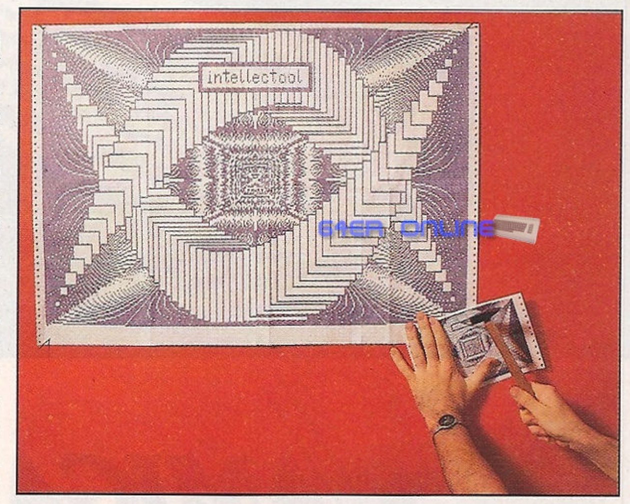
Seite 153



Play by Mail — Play by Modem

»Play by Mail«, das schriftliche Spiel über große Entfernungen hinweg, wird langsam durch eine neue Spielform ergänzt: »Play by Modem«. Was sich hinter beiden Begriffen verbirgt und wie man beides spielt, erfahren Sie in unserem Bericht. Wir zeigen Ihnen, wie man daran teilnimmt.
Seite 153

Seite 138



Test: Intellectool

Eine Basic-Erweiterung, die besonders auf das Floppy-Laufwerk und auf Drucker zugeschnitten ist. Die Erweiterung erlaubt eine mühelose Manipulation des Inhalts von Disketten. Für den Drucker können neben riesigen Hardcopies auch eigene Grafiken mit sehr hoher Auflösung ausgegeben werden.
Seite 138

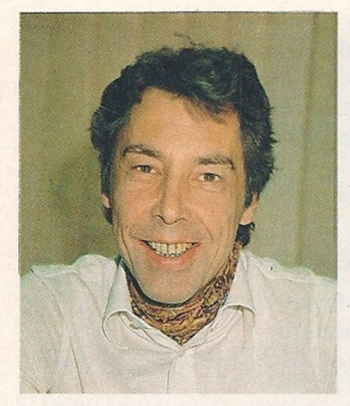
Seite 44



Versteht Sie Ihr Computer?

In einem Grundlagenartikel zeigen wir Ihnen, wie die verschiedenen Eingabegeräte eines Computers funktionieren. Welche Stärken und Schwächen haben sie? Tastaturen, Joysticks, Mäuse und Paddels sind die Schnittstellen Mensch-Computer, die wir unter die Lupe nehmen.
Seite 44





Reifezeugnis

Wenn im Lebenszyklus eines Produktes das Reifestadium erreicht ist, laufen Fertigung und Vertrieb problemlos, die Preise geraten unter Druck — und an dem Erzeugnis ist für Hersteller und Händler nichts mehr aufregend. Für den Benutzer wird das Produkt in diesem Stadium zum Teil erst richtig interessant, denn Kinderkrankheiten sind behoben, die Tücken des Objekts bekannt — und bei Computern wird das Softwareangebot attraktiv.

Für den C 64 gibt es neben Basic (samt den verschiedenen Erweiterungen) und Pascal jetzt — wir berichten darüber in dieser Ausgabe — beispielsweise auch sehr gute Versionen der Programmiersprachen C und Forth. Man kann damit ernsthaft lernen und arbeiten — im Gegensatz zu mancher »Probierversion« früherer Zeiten, die den Programmierer eher wieder zu Basic zurück als auf den Geschmack neuer Programmiersprachen brachte. Deutlicher Fortschritt ist auch auf einem anderen Gebiet zu beobachten: Textprogramme werden zugleich billiger und professioneller. Mit Star Texter kann man Standardaufgaben wie das Briefeschreiben nun wirklich schnell und komfortabel erledigen — bei bescheidenen 64 Mark Investitionskosten (für die Software).

Das Gefühl, den allerneuesten Computer zu haben, ist ja ganz schön — aber zu wissen, daß man für seinen Computer gute Software bekommt, ist besser.

Michael Pauly,
Redaktions-Direktor



AMIGA Der neue Supercomputer

Schon lange geistert ein Begriff durch die gesamte Computerwelt: Amiga. Computerfreaks träumen von diesem Commodore-Supercomputer, renommierte Fachzeitschriften veröffentlichten Gerüchte und Spekulationen über diese Maschine. Fantastische Fähigkeiten wurden dem Amiga angedichtet.

Am 23. Juli wurde der Amiga in New York endlich der Öffentlichkeit vorgestellt. Das 64'er-Magazin war für Sie dabei — und bringt Fakten statt Fiktionen. Doch diese Fakten belegen es eindeutig: Der Commodore-Amiga ist eine Traum-Maschine, ein neuer Schritt in der Evolution der persönlichen Computer.

Mit 4096 (!) Farben, einer Grafik-Auflösung von bis zu 400 x 640 Einzelpunkten, acht Sprites, hardwaregesteuerten Shapes und zwei getrennten, überlagerbaren Grafikbereichen werden völlig neue Maßstäbe hinsichtlich der grafischen Fähigkeiten von Computern gesetzt. Ein vierstimmiger Synthesizer produziert echten Stereo-Sound. Sprachausgabe und Digitalisierung analoger Eingaben (etwa vom an-

geschlossenen Plattenspieler oder Mikrofon) ist ebenfalls vorgesehen. Bei der Bearbeitung von Texten und bei der Programmierung kann der Anwender zwischen 40, 64 und 80 Zeichen Text pro Zeile wählen. Text und Grafik können beliebig gemischt werden. Eine ausgefeilte Window-Technik unterstützt die Benutzerführung und erleichtert die Programmierung.

Durch Multitasking können mehrere verschiedene Prozesse (vereinfacht gesagt: mehrere Programme) praktisch gleichzeitig aktiv sein. Beispielsweise kann man mit dem Amiga einen Brief schreiben, während zur selben Zeit »im Hintergrund« eine Datenbank neu sortiert und aktualisiert wird und zusätzlich die Einkommensteuererklärung ausgedruckt wird.

Der Commodore Amiga basiert auf dem 16-Bit-Mikroprozessor 68000 von Motorola, der mit einer Taktfrequenz von exakt 7,15909 MHz betrieben wird. In der Grundausbaustufe sind 256 KByte RAM vorhanden, die innerhalb des Gerätes auf 512 KByte aufgestockt werden können. Das Betriebssystem ist in 192 KByte ROM untergebracht und auf Flexibilität und Erweiterbarkeit durch zusätzliche Software ausgelegt. Der Hauptspeicher des Amiga ist extern auf bis zu 8,5 Megabyte (wahlweise RAM oder ROM) ausbaubar.

Ist die reine Aufzählung dieser Fähigkeiten und technischen Details allein schon beeindruckend, so hört sich die Beschreibung der damit wirklich möglichen Anwendungen wie die Realisierung eines Traums an. Insbesondere hinsichtlich Grafik und Animation wird eine völlig neue Leistungsklasse erreicht.

Neue Grafik-Dimensionen

Der Commodore Amiga verfügt über eine beeindruckende Palette von 4096 verschiedenen Farben, die allerdings nur in einem speziellen Modus gleichzeitig gezeigt werden können. Grundsätzlich wird der Bildschirminhalt auf der Basis einer Bit-Map (Bits im Speicher erscheinen als Grafikpunkte) angezeigt, ähnlich wie die hochauflösende Grafik des C 64. Ein spezieller Textmodus ist nicht vorhanden, alle Texte werden praktisch in die Bit-Map »gezeichnet«. Dadurch ergibt sich natürlich eine sehr große Flexibilität. Aussehen und Größe aller Zeichen lassen sich mit geringem Software-Aufwand ändern, selbst Proportionalschrift ist grundsätzlich möglich.

Für die Bit-Map, die beim Amiga als »Bit-Plane« bezeichnet wird, stehen zwei grundsätzliche Auflösungen zur Verfügung: 200 x 320 oder 200 x 640. Je nach gewähltem Modus werden dafür 8 beziehungsweise 16 KByte Speicher benötigt. Durch eine solche Bit-Map ist bereits einfarbige hochauflösende Grafik zu realisieren. Jedes gesetzte Bit entspricht dabei einem gesetzten Grafikpunkt.

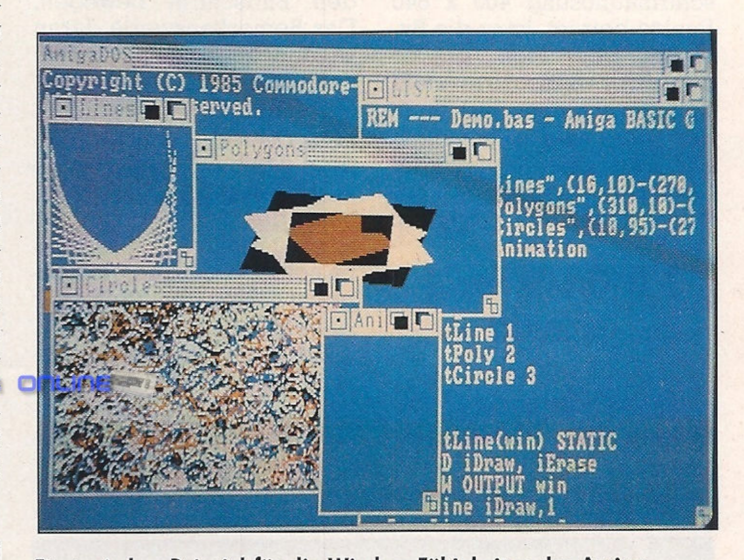
Bis zu 32 Farben pro Grafikpunkt

Der Amiga kann nun aber bis zu fünf solcher Bit-Maps oder -Planes überlagern. Nach wie vor wird die gleiche Anzahl an Grafikpunkten angesprochen, aber die korrespondierenden Bits in den überlagernden Bit-Planes kodieren jetzt eine Farbe für den entsprechenden Grafikpunkt. Dies geschieht auf dem Umweg über die Farbregister.

Es existieren 32 Farbregister, von denen jedes den Zahlencode für eine der 4096 Farben aufnehmen kann. Bei einer einzigen verwendeten Bit-Plane werden nur zwei Farbregister angesprochen: Register 0, wenn ein Bit gelöscht ist, und Register 1, wenn ein Bit gesetzt ist. Der in Register 0 gespeicherte Farbcode entspricht in diesem Fall der Hintergrundfarbe, während die Zeichenfarbe durch Register 1 bestimmt wird.

Werden zwei Bit-Planes verwendet, dann entspricht jedem Grafikpunkt eine Kombination aus zwei Bits, also können vier verschiedene Farbregister angesprochen werden. Das geht so weiter bis zu fünf Bit-Planes, mit denen dann alle 32 Farbregister angesprochen werden können. Das bedeutet, daß auch in hochauflösenden Modus (200 x 640) bis zu 32 Farben möglich sind. Allerdings bezahlt man diese Farbenpracht mit Speicherplatz: fünf Bit-Planes benötigen je nach gewählter Auflösung 40 oder 80 KByte RAM.

Doch damit sind die grafischen Fähigkeiten des Amiga noch lange nicht erschöpft, im Gegenteil, jetzt geht's erst richtig los.



Ein typisches Beispiel für die Window-Fähigkeiten des Amiga

Ein völlig neues Grafik-Gefühl: Computergrafik mit den Amiga

Im sogenannten »Interlace«-Modus kann sehr schnell zwischen zwei verschiedenen Bit-Planes hin- und hergeschaltet werden, wobei die Rasterpunkte der einen Bit-Plane geringfügig vertikal verschoben sind. Da dieser Wechsel 30mal in der Sekunde erfolgt, wird die Grafikauflösung durch diesen Trick rein optisch verdoppelt, nämlich auf 400 x 320 beziehungsweise 400 x 640 Punkte. Die Handhabung ist allerdings etwas komplizierter, denn jede durch Interlace erzeugte Grafik besteht in Wirklichkeit ja aus zwei Halbbildern, die in getrennten Speicherbereichen aufbewahrt werden.

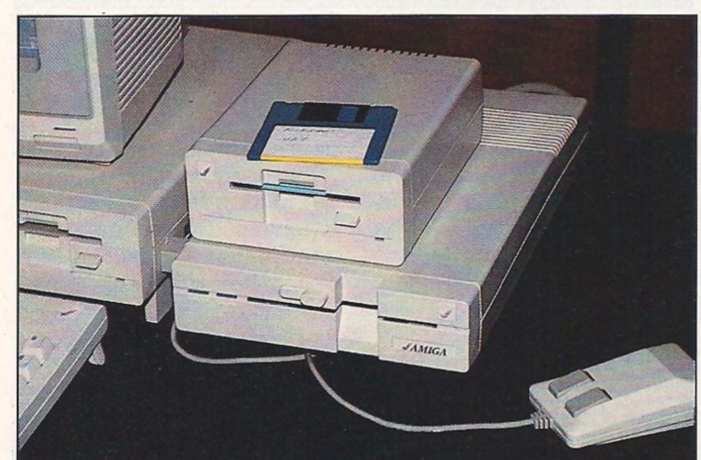
Wer es noch bunter mag, der kann mit dem »Hold-and-modify«-Modus wirklich alle 4096 Farben auf dem Bild-

schirm sichtbar machen. Allerdings ist dieser Modus nur mit Einschränkungen zu genießen: Sechs Bit-Planes sind dafür notwendig, von denen vier entweder ein Farbregister adressieren oder eine Modifikationsvorschrift für die Farbe des vorhergehenden Grafikpunktes darstellen. Gesteuert wird die Bedeutung dieser vier Bits durch die verbleibenden zwei. Die Farbe von Punkten ist also in diesem Modus nicht wirklich wahlfrei, sondern hängt in einem gewissen Rahmen von der Farbe des jeweils vorhergehenden Punktes ab.

Obwohl die maximale Bildschirmauflösung 400 x 640 Punkte beträgt, kann die Bit-Plane wesentlich größer sein. Der Bildschirm stellt dann nur einen Ausschnitt des gesamten »Playfields« dar. Spezielle Register im Video-Chip wählen den gewünschten Bildausschnitt an.

3 D-Grafik leicht gemacht

Außerdem ist es möglich, bis zu sechs Bitplanes in zwei Bildgruppen zu unterteilen, die verschiedene Bilder beinhalten. Die Bilder können gleichzeitig angezeigt werden, haben aber wählbare Prioritäten. So können komplexe 3D-Grafiken erzeugt werden, bei denen der Hintergrund durch den Vordergrund überdeckt wird.



Floppy-Laufwerke für 3½-Zoll- und 5¼-Zoll-Disketten. Rechts vorne die Amiga-Maus

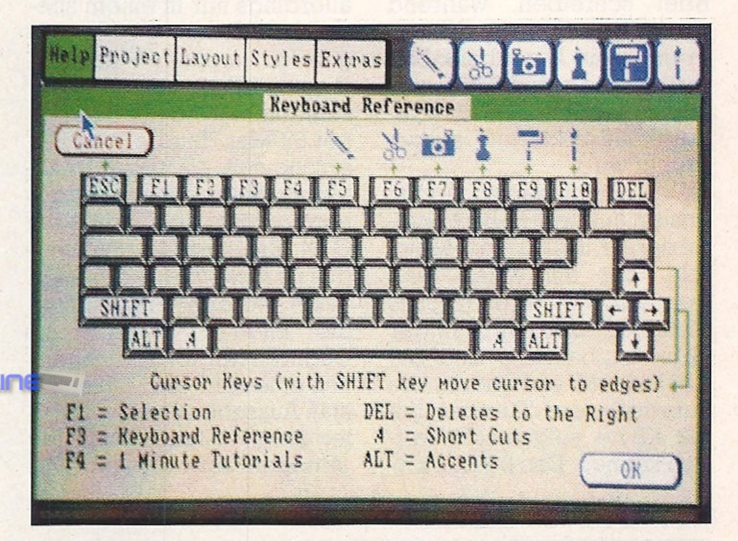
Sprites und Hardware-Shapes

Der Amiga kann ähnlich wie der C 64 hardwaremäßig acht Sprites erzeugen und steuern. Drei Farben plus Transparenz können für jeden Punkt des Sprites gewählt werden. Das allein bringt noch nicht viel Neues gegenüber dem C 64. Der Amiga verfügt aber neben diesen acht Sprites noch über eine bemerkenswerte Eigenschaft: Beliebige rechteckige Ausschnitte aus der Grafik-Bit-Map lassen sich als Shapes definieren und analog zu den Sprites über den Bildschirm bewegen. Das Bemerkenswerte daran ist, daß diese Shape-Animation nicht per Software (also letztendlich vom 68000-Prozessor) ausgeführt wird, sondern daß dafür eine spezielle Hardware-Einheit innerhalb des Video-Chips zuständig ist. Dadurch wird eine hohe Geschwindigkeit erreicht, ohne den Prozessor zu belasten.

Zur komfortablen Steuerung von Sprites und Shapes stehen spezielle Betriebssystem-Routinen zur Verfügung, die die Animation dieser Objekte übernehmen. Positionen und das Aussehen von Sprites und Shapes können so kontinuierlich geändert werden, wodurch sich Trickfilm-Effekte ergeben. Der Anwender braucht sich um Einzelheiten nicht zu kümmern. Nachdem er dem System mitgeteilt hat, wo die Bildfolgen im Speicher liegen und Reihenfolge und Bewegungsablauf festgelegt wurde, geschieht alles weitere automatisch, ohne Programmsteuerung.

Stereo-Synthesizer

Ein eingebauter vierstimmiger Synthesizer macht aus dem Amiga ein fast perfektes Musik-Gerät. Damit man den Sound auch in optimaler Qualität genießen kann, sind Anschlüsse für die Stereo-Anlage vorhanden.



Der Amiga-Tutor von Mindscape erleichtert die Programmierung des Amiga

Interessant ist auch die Möglichkeit, Töne über die Audio-in-Buchsen in den Computer zu bringen, dort zu digitalisieren und als Tonfile abzulegen. Der Originalton kann jetzt beliebig manipuliert und verändert werden.

Floppy-Laufwerk eingebaut

Interessant ist auch ein eingebundener vollwertiger Sprachsynthesizer, dessen Verständlichkeit weit über die bisher bekannten Systeme reicht. Der Wortschatz ist unbegrenzt. Es ist einfach nur der auszugebende Text einzutippen, und schon legt der Amiga mit einer nach Wahl männlichen oder weiblichen, lebendig klingenden Stimme los.

Fest in den Amiga integriert ist ein 3½-Zoll-Disketten-Laufwerk mit einer Speicherkapazität von 880 KByte (formatiert). Weitere drei Diskettenlaufwerke (wahlweise 3½ oder 5¼ Zoll) können angeschlossen werden.

Zur Verbindung mit der Außenwelt dient ein programmierbarer paralleler Port, der von der Software normalerweise als Centronics-Schnittstelle angesprochen wird, ferner eine RS232C-Schnittstelle mit einer maximalen Übertragungsrate von 500000 bps (Bits pro Sekunde). Daneben sind vorhanden ein TV-Anschluß, ein Video-Composite-Anschluß und ein RGB-Anschluß (sowohl digital wie beim IBM-PC als auch analog), zwei Stereo-Audio-Ausgänge, ein Expansions-Port, an dem der komplette Adreß- und Datenbus des 68000 herausgeführt ist, ein Anschluß für die Tastatur und schließlich noch zwei programmierbare Control-Ports für Maus, Joystick, Lichtgriffel oder Grafiktablett.

Der Amiga kostete in den USA 1295 Dollar (Grundausstattung mit 256 KByte RAM und einem Disketten-Laufwerk, aber ohne Monitor). Der Amiga-Monitor kostet nochmals zirka 500 Dollar.

Ganz auf der Commodore-Linie, beruht die Leistungsfähigkeit des Amiga im wesentlichen auf dem Zusammenwirken modernster

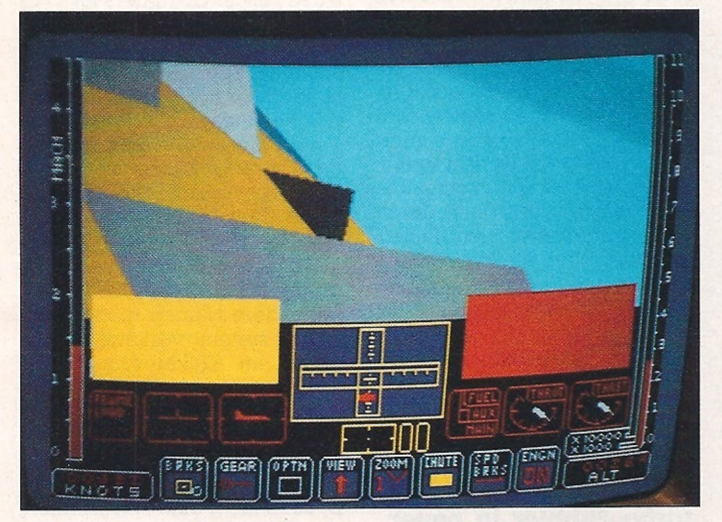
Hardware-Komponenten:

Der 68000-Prozessor wird in seiner Arbeit wesentlich von drei speziellen »Custom Chips« unterstützt. Der eine ist für die gesamte Grafik verantwortlich, der zweite für die Animation, der dritte für den Sound, I/O-Operationen und Diskettenunterstützung.

Eine wesentliche Steigerung des Datendurchsatzes und damit der Effizienz des Systems wurde erreicht durch konsequente Arbeitsteilung zwischen dem Prozessor und den Hilfsbausteinen, die man besser schon fast als Coprozessoren bezeichnen sollte.

Geschwindigkeit durch DMA

Die Peripheriebausteine greifen selbständig per DMA (Direct Memory Access, direkter Speicherzugriff) auf den Arbeitsspeicher zu und holen sich so die benötigten Parameter oder Daten, ohne daß der Prozessor wertvolle Rechenzeit damit vergeuden müßte, diese Parameter selbst zu holen und dann in irgendwelche Register zu schreiben.



Radar-Raiders von Sublogic — ein schneller Flugsimulator für den Amiga

Daten werden so auch von einem Baustein direkt in einen anderen übertragen, ohne daß der Umweg über den Prozessor notwendig wäre. Auch Peripherie-Geräte können mittels DMA direkt in den Hauptspeicher des Amiga schreiben oder Daten daraus lesen. Wenn insgesamt vier Chips (einschließlich Prozessor) direkt auf den Speicher zugreifen können, dann erfordert das natürlich ein exaktes Timing, denn es ist ja nur je ein Adressen- und Datenbus vorhanden. Während bestimmter Taktzyklen ist daher der Bus für den Prozessor gesperrt, was dem aber nicht viel ausmacht, da diese Zyklen so gewählt wurden, daß der 68000 dann gerade einen internen Maschinenzyklus bearbeitet und nicht auf den Speicher zugreifen muß.

Kritisch wird es allenfalls dann, wenn viele Shapes und Sprites bewegt werden müssen. Der dafür zuständige Prozessor »stiehlt« dem 68000 dann einfach einige (oder auch viele) Taktzyklen: Der Prozessor wird langsamer, obwohl er gar nichts Zusätzliches zu tun hat.

Im übrigen läuft die gesamte notwendige Kommunikation zwischen dem Prozessor und anderen Chips ausschließlich über Interrupt. Der 68000 muß also vereinfacht gesagt nicht regelmäßig überall herumfragen, ob irgendwas passiert ist, sondern nur dann, wenn etwas passiert ist, wird er einfach unterbrochen, um sofort darauf zu reagieren.

Zusammen mit der DMA-Fähigkeit der »Custom Chips« verleiht dies dem Amiga seine schon fast unheimliche Geschwindigkeit, die sich insbesondere bei der Grafik bemerkbar macht. Aber auch seine Multitasking-Fähigkeit beruht auf einer ausgefeilten Interrupt-Steuerung. Multitasking bedeutet nichts anderes, als daß die Ausführung mehrerer Programme zur gleichen Zeit möglich ist. Ein Beispiel: Sie lassen einen langen Text ausdrucken, führen währenddessen Neukalkulationen in Ihrem Tabellenkalkulationsprogramm aus und hören eigene Kompositionen aus dem Amiga-Synthesizer. Und all das können Sie sich auch noch auf dem Bildschirm in verschiedenen Fenstern (Windows) ansehen.

Bereits mitgeliefert wird an Software AmigaDOS, Voice Synthesis Library, ABasic (von Microsoft), Tutorial (von Mindscape) und Kaleidoscope (von ECA).

Im ROM ist das Betriebssystem und die »Benutzeroberfläche« Intuition eingebettet. Intuition ähnelt mit der grafischen Darstellung aller Funktionen und den »Pull Down«-Menüs (Textfenster, die sich öffnen, wenn sie mit der Maus »angeklickt« werden) dem vom Atari 520 her bekannten GEM und der Benutzeroberfläche von Apple's Macintosh, ist allerdings wesentlich komfortabler und multitasking-fähig. In verschiedenen Textfenstern können so unterschiedliche Programme ablaufen und bearbeitet werden — und alles gleichzeitig.

Video-Bilder direkt bearbeiten

Die externe Einspeisung von verschiedenen Video-Quellen (Kamera, Recorder, CD-Player etc.) sowie die Digitalisierung und Weiterverarbeitung im Computer sind mittels Framegrabber und Genlock-Interface einfach möglich.

Für den Amiga soll es bereits zum Zeitpunkt seines Verkaufs im September in den USA ein 20-MByte-Festplattenlaufwerk, ein 20-MByte-Tape-Backup und ein 2400-bps-Modem von Tecmar geben. An Sprachen wird dann auch C, Turbo Pascal, Basic, Logo, Lisp und Assembler verfügbar sein.

Zur Zeit sind viele Firmen dabei, altbekannte Computerspiele für den Amiga umzuschreiben oder neue Software dafür zu entwickeln. So ist der Amiga auf dem besten Wege, eine neue Generation von Computerspielen mit bisher unbekannter Komplexität, Grafik und Geschwindigkeit ins Leben zu rufen.

Amiga als IBM-PC

Die größte Überraschung während der Präsentation des Amiga in New York war allerdings die problemlose Abarbeitung von Programmen, die unter MS-DOS für den IBM-PC und kompatible geschrieben wurden. Per Software wird der 68000-Prozessor im Amiga in die Lage versetzt, beispielsweise Lotus 1-2-3, dBase III oder Wordstar unter MS-DOS (sowohl mit den 5¼- und 3½-Zoll-Disketten) zu verarbeiten. Dazu emuliert (simuliert) der 68000 einen 8088-Prozessor in MS-DOS-Umgebung, wobei keine Geschwindigkeitseinbußen gegenüber dem Original-IBM bemerkt wurden. Damit entpuppt sich der Amiga nicht nur als Grafik- und Spielewunder, sondern zeigt auch seine Stärken im kommerziellen Bereich.

Fazit

Im Amiga sind derartig viele außergewöhnliche Eigenschaften integriert, daß eine abschließende Bewertung sicherlich nicht endgültig sein kann. Marshall F. Smith, President und Chief Executive Officer, sieht das Einsatzgebiet des Amiga sowohl im »Home-, Office- und School-«Bereich.

Es läßt sich wirklich sagen, daß der Amiga in nahezu allen Bereichen seinen Mann (Frau) stehen wird. Seine ganz besonderen Stärken sind sicher Grafik und Animation. Ihn deshalb als reine »Spielmaschine« zu bezeichnen, wäre dennoch verfehlt, denn eigentlich gehören grafische Darstellungen in fast jedes kommerzielle Programm. Alles in allem eine universelle Maschine für jedermann, der bereit ist, um die 5000 Mark für den Spaß und mit dem am Computer auszugeben. (aa/ev)

ProLogic DOS lädt 25mal schneller!

ProLogic DOS ist der Name eines neuen Erweiterungs-Kit für den C 64 und sein 1541-Laufwerk. Das neue DOS wird, wie von Speeddos und Turbo Access bekannt, eingebaut. Danach stehen unter anderem folgende Funktionen zur Verfügung:
— parallele Datenübertragung
— 25mal schneller laden
— 13mal schneller speichern
— 10mal schnellerer Datenzugriff beim Laden (achtmal beim Speichern) von sequientiellen oder relativen Dateien
— Funktionstastenbelegung
— RENEW-Funktion
— korrigiertes LIST (Steuerzeichen)
— korrektes SAVE & REPLACE
— Anzeige des Speicherbereichs beim Laden
— Laden des ersten Programms von Disk
— vereinfachte DOS-Befehle
— Centronics-Schnittstelle
— freier User-Port
— freier Expansion-Port (durchgeführt)
— kürzerer RAM-Test
— RS232 voll erhalten
— schnelles Formatieren
— schnellere Kopfbewegungen
— automatisches Zentrieren der Diskette beim Einlegen
— vermindertes Anschlagen beim Formatieren oder Fehlern
— 35 oder 40 Tracks; umschaltbar

Wir hatten Gelegenheit, ProLogic DOS einem Kurztest zu unterziehen. Außer einigen Programmen aus Düsseldorf gab es nichts, das ProLogic DOS vom korrekten Laden abhalten konnte. Ein 88 Blöcke langes Programm wurde beispielsweise in sensationellen zwei Sekunden geladen. Besonders beeindruckend ist das Laden von Grafikbildern, sie laufen nicht wie gewohnt ins Bild, sondern erscheinen fast schlagartig. Durch die automatische Umschaltung auf 40 Tracks werden die Disketten besser ausgenutzt, statt der üblichen 664 Blöcke stehen nun 749 Blöcke zum Speichern von Programmen zur Verfügung. Normal beschriebene Disketten werden natürlich weiterhin gelesen. ProLogic DOS soll ab Anfang September zu einem Preis von 298 Mark erhältlich sein.

Info: Jann Datentechnik, Glimmerweg 22, 1000 Berlin 47, Tel. (030) 731184

Btx: Der EUROM funktioniert endlich

Der EUROM-Chip von Valvo, dem im Artikel über die Hannover Messe in der Ausgabe 7/85 die Schuld an den noch nicht vorhandenen, preiswerten Decoder für Btx gegeben wurde, soll nach Aussage von Valvo jetzt fertig sein. Das heißt, die seit Anfang 1985 in Fertigung befindliche Version arbeitet in allen Parametern fehlerfrei. Diese Version wird zum heutigen Zeitpunkt an die Geräteindustrie ausgeliefert.

Der EUROM-IC 5350 soll alle Punkte der seit 1982 festgeschriebenen Spezifikationen erfüllen. Mit diesem EUROM-IC soll nach Aussage von Valvo immer noch die preisgünstigste Konzeption für einen Btx-Decoder realisierbar sein. Die angesprochene Wärmeempfindlichkeit konnte vermindert werden, der IC arbeitet jetzt in einem Temperaturbereich von 0 bis 60 Grad einwandfrei. Damit ist von Valvo endlich die Voraussetzung für die Gerätevielfalt auf dem Markt der Btx-Endgeräte geschaffen worden. (aa)

Profi-Pascal nicht neu

Gegendarstellung zum Artikel »Pascal für Profis«, 64'er, Ausgabe 8/85, Seite 122.

Im Artikel wird der Eindruck erweckt, »Profi-Pascal« sei eine Data-Becker-Entwicklung (»Im zweiten Anlauf hat Data-Becker es geschafft: ...«).

Richtig ist, daß 64'er bereits Anfang 1984 den hier beschriebenen Compiler unter dem Namen »phs-Pascal« ausgiebig testete und nicht darüber berichtete.

Das unter dem Namen »Profi-Pascal« beschriebene System ist mit dem seit 1980 von unserer Firma für die Entwickler Schnepf und Tackmann vertriebenen »phs-schtac-pascal« nahezu identisch. Für »Profi-Pascal« sind nur unwesentliche Änderungen vorgenommen worden. Data-Becker bringt also kein »neues Programmpaket«, sondern lediglich eine in der Optik geschönte und durch Turbo-Routinen verschnellerte Version des »phs-schtac-pascal v5.0« auf den Markt.

1980 schloß einer der Autoren einen Exclusiv-Vertriebs-Vertrag mit unserem Hause und zeichnete auch im Copyright-Vermerk mit »Copyright H. Schnepf & D. Tackmann, West Germany 1982, Technische Fragen und Vertrieb phs EDV Beratung«. Die von uns seit 1982/83 vertriebene Version 6 des Pascal-Compilers für C 64 wurde noch mit Copyright H. Schnepf & D. Tackmann 1982«, später »Copyright H. Schnepf & D. Tackmann 1984« gezeichnet.

Schließlich ist sowohl dem Verlag Markt & Technik als auch der Firma Data-Becker das System spätestens seit 1983 bekannt. Zum Zeitpunkt Ihrer Veröffentlichung waren bereits mehr als 500 Compiler (einschließlich der Version für C 64) unter dem Namen »phs-schtac-pascal-system« verkauft, wohingegen im Artikel vom »neuen Profi-Pascal« gesprochen wird.

Es bleibt festzuhalten, daß der »Profi-Pascal«-Compiler
a) kein von Data-Becker entwickeltes Programm ist
b) zumindest eine Nachbildung des durch Tackmann/Schnepf erstellten und über phs vertriebenen Produktes »phs-schtac-pascal v6« darstellt.
c) mit einem Handbuch ausgeliefert wird, das wegen der Identität von Befehlsstruktur, Sprachumfang und -erweiterungen auch für den Umgang mit »phs-schtac-pascal v6« eingesetzt werden kann.

(M. Penzkofer, phs)

Nicht nur bei »Lacoste«

Es hat sich herumgesprochen, daß hochwertige Konsum-, aber auch Investitionsgüter kopiert werden. Das Spektrum der »Nachahmungen« reicht von einfachen Cartier-Feuerzeugen bis hin zum hochwertigen Maschinenteil. Dieses Unwesen hat sich auch in der Computerbranche breitgemacht. Abgesehen von den reichlich diskutierten »Softwarepiraten« gibt es noch eine andere, nicht minder aktive Gruppe — die »Hardware-Kopierer«. Vom ganzen Computer (Apple), bis zur einfachen Einsteckplatine, wird fast alles kopiert. Auch im Rahmen unserer redaktionellen Tätigkeit sind wir schon so manches Mal auf Geräte gestoßen, deren Herkunft zumindest zweifelhaft war. Manchmal sind es nur Teile eines Gerätes, die kopiert und in eigene oder gekaufte Entwicklungen eingebaut werden, manchmal sind es genaue Kopien des Originals. Besonderer Beliebtheit bei den Kopierern erfreut sich die Firmware (auf ROMs oder EPROMs gespeicherte Software). So wurde von uns beispielsweise ein Drucker getestet, dessen Fähigkeiten auffallend mit denen eines Konkurrenzproduktes übereinstimmten (der in der nächsten Ausgabe vorgestellt wird, und das eigentliche Original ist). Nun ist es natürlich recht problematisch festzustellen, wer denn eigentlich der Urheber einer Entwicklung ist, denn gerade in der Computerbranche werden viele Lizenzen vergeben. Oft hilft da einfaches Fragen nicht mehr. So geschehen bei unserem EPROMmer-Vergleichstest. Zwei der uns zum Test vorgelegten Geräte konnten, durch Vergleich der Bauteile und der Software, eindeutig als Nachahmungen identifiziert werden, gleichwohl beide Hersteller behaupteten, das Gerät selbst entwickelt zu haben. Den besten Beweis für unsere Annahme lieferten aber die Geräte selbst. In beiden Fällen war es möglich, das Originalgerät mit der Treibersoftware der Kopie zu betreiben und umgekehrt — diese Tatsache spricht wohl für sich. Den eigentlichen Hersteller der Entwicklungen konnten wir in diesen Fällen nur deshalb ermitteln, weil wir zuverlässig wußten, welches Gerät zuerst auf dem Markt war. Dies mag nur die Spitze eines Eisberges sein. Viele der uns eingesendeten Modulplatinen unterscheiden sich in fast keinem Punkt. Auch ist es äußerst schwer festzustellen, wo unberechtigt »Know-how« eines anderen verwendet, aber eben geschickt versteckt wurde. Manchmal sind die »geklauten« Teile eines Produktes auch verschwindend klein, aber für die Funktionalität von existenzieller Wichtigkeit. So verwenden beispielsweise viele der derzeit verkauften Programme oder Steckmodule zur Beschleunigung des Ladevorgangs, die (möglicherweise etwas veränderten) Routinen unseres Hypra-Load. Dies sind Bereiche in denen der Nachweis einer Aneignung fremden geistigen Eigentums fast nur noch mit einem extrem hohen Aufwand zu führen ist. Erschwerend wirkt sich in diesem Zusammenhang, immer noch die rechtlich unklare Abgrenzung zwischen, durch technische Zwänge nicht anders zu realisierenden Algorithmen und einmaligen geistigen Leistungen aus. Wir werden auch weiterhin versuchen, sowohl im Interesse unserer Leser als auch dem der vielen ehrlichen Konstrukteure, nur solche Geräte in unserer Zeitschrift vorzustellen, bei denen es keinen Zweifel am Urheberrecht gibt. (aw)

Turbo-Nibbler in neuer Version

In der letzten Ausgabe berichteten wir über das Kopierprogramm »Turbo Nibbler« und sagten, daß nicht alle Lesefehler kopiert werden könnten. Inzwischen ist die verbesserte Version »Turbo Nibbler 2.1« erhältlich, die jetzt alle Lesefehler und auch Killertracks auf den Spuren 1 bis 41 kopiert. (bs)

Info: Eurosystems, Verlengde Parkweg 6, 6717 gn Ede, Holland





C 128 C 128 C 128 C 128 C 128

Neu: Das 128'er Magazin

Für alle, die sich von Anfang an für den neuen Commodore 128 interessieren, gibt es ab der nächsten Ausgabe einen großen Sonderteil im 64'er. Hier finden Sie die heißesten Informationen und jede Menge Tips und Tricks rund um den C 128. Ausgabe 10 ist am 20. September überall im Zeitschriftenhandel erhältlich.

Wir suchen die ersten 128'er Profis

Der neue C 128 kommt in diesen Tagen in den Handel. Vielleicht gehören Sie ja zu den ersten Besitzern dieses vielseitigen und leistungsfähigen neuen Computers und haben schon erste, tiefergehende Erfahrungen damit gesammelt oder interessante Programme dafür geschrieben?

In diesem Fall sollten Sie Ihre Erfahrungen nicht für sich behalten. Tragen Sie doch einfach mal zusammen, was Sie über den C 128 herausgefunden haben, das nicht im Handbuch steht. Wir werden alle guten Tips und Tricks zu diesem Computer sowie die besten Programme dafür im 64'er-Magazin veröffentlichen.

Bitte vermerken Sie bei allen Zusendungen, mit welcher Gerätekonfiguration Sie arbeiten und ob sich Ihr Beitrag auf den 128-Modus oder auf den CP/M-Modus bezieht. Beiträge für den C 64-Modus schicken Sie bitte nicht unter dem Stichwort C 128, sondern als normale C 64-Programme oder Tips ein. Bei Programmeinsendungen legen Sie bitte unbedingt eine Diskette/Kassette mit Ihrem Programm sowie eine möglichst ausführliche Beschreibung bei. Selbstverständlich werden alle abgedruckten Beiträge angemessen honoriert — es winken bis zu 2000 Mark, wenn Ihr Programm Listing des Monats wird.

Schicken Sie Ihre Tips und Programme an

Markt & Technik
Verlag AG
Redaktion 64'er
Kennwort: C 128
Hans-Pinsel-Str. 2
8013 Haar bei München

C 128: Professionelle CP/M-Software für 199 Mark

Gleich drei professionelle CP/M-Programme für den Commodore 128 und die Schneider-Computer bietet der Markt & Technik Verlag an: Wordstar 3.0 (Textverarbeitung) mit Mailmerge (Serienbriefe), dBase II (Datenbanksystem) und Multiplan (Tabellenkalkulation).

Alle drei Programme bieten eine Centronics-Druckerschnittstelle über den User Port sowie selbstverständlich den deutschen Zeichensatz. Natürlich sind auch die integrierten Hilfstexte ebenfalls in Deutsch.

Wordstar arbeitet mit voll bildschirmorientierter Formatierung, das heißt der Text erscheint so am Bildschirm, wie er später ausgedruckt wird. Textbausteine, Blockverschieben, Suchen und Ersetzen, Serienbriefe und jede Menge Formatierungsmöglichkeiten sind selbstverständlich.

dBase II ist ein relationales Datenbanksystem, das bis zu 65000 Datensätze rational und anwenderfreundlich verwalten kann. Dank der integrierten Datenbank-Programmiersprache sind individuelle Anwendungen sehr leicht und übersichtlich zu programmieren.

Multiplan schließlich ist eine leistungsfähige Tabellenkalkulation. Planung, Kalkulation, Prognosen, Abrechnungen — Multiplan ist immer dann geeignet, wenn umfangreiches Zahlenmaterial ausgewertet werden muß.

Diese Programme wurden für den professionellen CP/M-Anwender entwickelt und werden seit Jahren ständig verbessert. Den Preis von nur 199 Mark (!) für jedes Programmpaket kann man nur als Sensation bezeichnen — schlechte Zeiten für 16-Bit-PCs, für die die gleichen Programme unter MS-DOS zwischen 770 und 1200 Mark kosten — und das bei gleicher Leistung! Denn es wird vermutlich nur wenigen Anwendern einleuchten, warum sie für einen Personal Computer mit vergleichbarer Leistung wie der C 128 hinsichtlich Grafik, Farbe und Sound fünfstellige Summen hinblättern müssen, nur damit die Software dann — bei gleicher Leistung — auch noch drei- bis sechsmal so teuer ist. Wordstar, dBase II und Multiplan — das sind jedenfalls drei weitere gute Gründe für den Commodore 128. (ev)

Wordstar, dBase II und Multiplan sind Originalprodukte der Hersteller MicroPro (Wordstar), Ashton Tate (dBase) und Microsoft (Multiplan). Alleinvertrieb in Deutschland für Commodore 128 und Schneider-Computer durch Markt & Technik Verlag AG, Hans-Pinsel-Str. 2, 8013 Haar bei München, Preis pro Software-Paket (inklusive Programmdiskette, ausführliches Handbuch, kompakte Befehlsübersicht) 199 Mark.

Fragen und Antworten zum C 128

Ist der PC 128 im C 64-Modus wirklich voll kompatibel?

Nach wochenlangen Tests mit allen möglichen Programmen, mit Datasette, 1541-Floppy und C 64-Steckmodulen können wir guten Gewissens versichern: Der C 128 ist zu 99,99 Prozent zum C 64 kompatibel. Lediglich Programme, die ganz exotische illegale Opcodes (undefinierte Maschinensprache-Befehle des Prozessors, die je nach Prozessor-Bauserie verschiedene Wirkungen haben können) verwenden, laufen auf dem C 128 nicht unbedingt. Allerdings sind derartige Programme auch nicht auf allen C 64-Computern lauffähig, so daß man nicht von einer Einschränkung der Kompatibilität sprechen kann. Im übrigen sind solche Programme meist von Privatpersonen für den eigenen Gebrauch geschrieben und nicht im freien Handel erhältlich.

In England wurde kürzlich ein spezielles Schutzverfahren für

C 128 C 128 C 128 C 128 C 128 C 128 C 128 C 128 C 128 C 12

Fünfte Commodore-Fachaustellung in Frankfurt

Mit etwa 100 Ausstellern und gut 30000 Besuchern rechnen Insider der Frankfurter Messeszene auf der »Fünften Internationalen Commodore-Fachausstellung« (CFA). Zwischen dem 18. und 20. Oktober 1985 werden in der Main-Metropole alle aktuellen Produkte aus der Welt des Mikro-Matadors zu sehen sein. Außerdem stehen Workshops und Seminare, aber auch Spaß und Spiel auf dem Programm.

Tasächlich deuten alle Anzeichen darauf hin, daß der Aufwärtstrend, der die Commodore-Fachausstellung CFA seit fünf Jahren kennzeichnet, ungewöhnliche Ausmaße erreicht. Denn: seit der Präsentation und Einführung der neuen und attraktiven Home-, Hobby-, Semiprofi- und Personal Computer registriert der Kleinrechner-Konzern (Commodore) einen kräftigen Nachfrageschub.

Vor allem wurde der PC 10/20 ein Volltreffer: Zwischen Februar und August dieses Jahres flossen etwa 10000 Geräte in den Markt. Konstatiert Harald Speyer, Vice President von Commodore International Ltd.: »Die uneingeschränkte IBM-Hard- und Softwarekompatibilität unseres PC sowie das sensationelle Preis-/Leistungsverhältnis programmierten diesen Computer zwar von vornherein zum Renner. Trotzdem übertraf sein Erfolg alle Prognosen.«

Einen guten Start hatten inzwischen zudem der C 128 (C/PM-Standard und C 64-kompatibel) sowie das kommerzielle System CBM 900 (Unix-kompatibel). Weiterhin als Dauerbrenner erweist sich derweil der C 64 — er wurde in der Bundesrepublik bislang nahezu 800000mal und weltweit gar rund fünfmillionenmal abgesetzt.

Der Commodore-Boom wird von einer wachsenden Popularität getragen: Gemäß Erhebungen der Trenddetektive von »Basisresearch« (Frankfurt) verdreifachte sich der sogenannte »gestützte« Bekanntheitsgrad des Mikro-Champions innerhalb nur eines Jahres und erreichte im Dezember 1984 bei der Gesamtbevölkerung 19 Prozent (nach IBM mit 38 und Nixdorf mit 22 Prozent). Personen, die sich schon mit Kleincomputern befaßt hatten, nannten bei der Frage »Wer baut die leistungsfähigsten Heimrechner?« zu 34 Prozent Commodore (vor IBM mit 24 und Apple mit 13 Prozent).

Auf der nächsten CFA wird Commodore die Nettoausstellungsfläche der Halle 1 des Messegeländes von »Mainhattan« diesmal zum größten Teil belegen.

Mehrmals täglich läuft eine informative Multivisions-Show, die das derzeitige Produkt-Spektrum auf eingängige Weise visualisiert. Der Schüler mag ausprobieren, wie sich mit spielerisch gestalteten Programmen leichter Deutsch, Mathematik oder Fremdsprachen erlernen lassen. Der aufstiegsbegierige Karrierist entdeckt, wie er sein persönliches Arbeitspensum effizienter meistert. Dem Ingenieur öffnen sich die Perspektiven des Computer Aided Design (CAD) via Mikro.

Die kommerziellen Fachbesucher dürfte das reichhaltige Bündel branchenspezifischer Lösungen ansprechen. So sind komplette und speziell zugeschnittene Programme für alle Handwerkszweige des Baugewerbes zu erwarten. Weitere typische Applikations-Beispiele: Ärzte, Zahnärzte, Versicherungen, Autobahntankstellen, Getränkehandel, Labors und Heizungs- beziehungsweise Sanitärwesen. Praxisnahe Software gibt's indes nicht nur für den kaufmännischen Sektor, sondern ebenso auch für technische Ressorts wie die Fertigungssteuerung. Hinzu kommen Produkte, mit denen die Netz- und Kommunikationsfähigkeit der Commodore-Maschinen voll genutzt werden können.

Außer den Software-Schöpfern werden auch die Hersteller von Zusatzgeräten (zum Beispiel Drucker), Datenträger (zum Beispiel Disketten) und Zubehör (zum Beispiel Farbbänder) sowie Fachverlage anwesend sein. Commodore wird weitere Folgen seiner Mikro-Sachbuchreihe vorstellen.

Besonders die jungen Freaks werden voll auf ihre Kosten kommen. Für sie gibt es eine »Spielnische«, wo sie ihrer Phantasie mittels zahlreicher Play-Programme freien Lauf lassen können. Für die Fans hält eine Commodore-Boutique begehrte Populär-Artikel bereit: Aktenkoffer, Taschenrechner, Sportjacken, Schirme und andere reizvolle Gegenstände mit dem Commodore-Signet. Und natürlich auch spezielle Utensilien — zum Beispiel Trikots — des Werbepartners 1. FC Bayern München.

Um die Stimmung während der drei tollen Mikro-Tage hochzuhalten, ließen sich die Marketing- und Messestrategen des Konzerns noch weitere Gags einfallen: Ein Funk-Moderator wird täglich Commodore-Computer verlosen. Voraussichtlich reist auch der Mikro-Guru Jim Butterfield aus Kanada an, um im Kreise seiner Jünger neueste Erkenntnisse über den »Mikro-Kosmos« auszuplaudern. (aa)

Die Musik-Hardware zum C 64

Der C 64 verfügt bekanntlich über nicht zu verachtende musikalische Möglichkeiten. Doch wie nutzt man diese Fähigkeiten optimal?

In letzter Zeit findet man in den Fachzeitschriften immer häufiger Werbung für an den C 64 anschließbare Keyboards, oder auch für Mini-Klaviaturen, die man einfach auf die Tastatur aufsetzen kann. Wir haben uns diese Geräte näher angesehen, um Sinn und Zweck solchen Zubehörs zu erkunden.

Keyboards

Das wohl bekannteste derartige Keyboard ist das der Firma Wersi. Es wird über den Expansionport mit dem Computer verbunden. Durch die mitgelieferte Software läßt sich wahlweise ein einstimmiger Synthesizer oder ein polyphones (dreistimmiges) Keyboard realisieren. Ähnliche Geräte in der gleichen Preisklasse werden von verschiedenen Firmen vertrieben (siehe Info). Der Vorteil dieser Art von Klaviatur liegt auf der Hand: Sie ist über mehrere Oktaven hinweg so zu spielen, als hätte man ein normales Keyboard vor sich. Im Gegensatz zur Tastatur, bei der man schon ganz schön geübt sein muß, um wenigstens einstimmige Melodien reibungslos spielen zu können. Störend ist allerdings der Umstand, daß nur maximal drei Töne gleichzeitig angeschlagen werden können, denn wenn man es beispielsweise gewohnt ist, auf konventionellen Keyboards mindestens vierstimmig zu spielen, klingt das Ganze auf dem C 64 ziemlich schräg. Aber dieses Problem ist eben hardwarebedingt und läßt sich nicht aus der Welt schaffen.

Das Wersiboard

Der Tonumfang beim Wersiboard beträgt bei 49 Tasten vier Oktaven. Es ist im Commodore-Design gehalten und die Verarbeitung macht einen soliden Eindruck. Die beiden mitgelieferten Programme konnten allerdings nicht ganz überzeugen, da sie wirklich nur das allernötigste an Benutzerfreundlichkeit bieten. Es lassen sich zwar beim monophonen Synthesizer alle gängigen Einstellungen vornehmen und auch beeindruckende Effekte erzielen, aber gerade beim polyphonen Programm hätten sich die Programmierer mehr Mühe geben müssen. So kann man leider nur eine gemeinsame Klangfarbe für alle drei Stimmen einstellen. Es wäre dabei doch sehr vorteilhaft, wenn man bestimmten Oktaven einzelne Sounds zuordnen könnte und sich so Baßbegleitung und Melodie mehr von einander absetzen würden. Es wäre wünschenswert, wenn zu diesem ansonsten ausgezeichneten Keyboard auch ein wirklich professionelles Synthesizer-Programm angeboten werden würde.

Das Wersiboard kostet 495 Mark; die Software ist im Preis enthalten. Die Anschaffung eines derartigen Keyboards will bei einem Preis, der fast an den des Computers selbst heranreicht, natürlich gut überlegt sein. Wer ernsthafter die musikalischen Fähigkeiten seines C 64 ausnutzen will und vor allem selbst spielen möchte, für den ist ein solches Keyboard natürlich unverzichtbar. Wer aber nur ab und zu etwas »herumklimpern« will, sollte sich wohl doch nach preiswerteren Lösungen umschauen. Auch wenn man eher ans Programmieren von Musikstücken und Soundeffekten als ans selber spielen denkt, ist die Anschaffung eines leistungsfähigen Composer-Programms wohl sinnvoller, als das Geld für ein Keyboard auszugeben.

Aufsatz-Klaviaturen

Eine auf den ersten Blick weniger sinnvolle Erweiterung scheint eine Aufsatz-Klaviatur zu sein. Eine unten offene Kunststoffkonstruktion wird auf die Tastatur aufgesetzt und bei niedergedrückter Taste überträgt die Klaviatur diese Bewegung auf eine entsprechende Rechnertaste. Die Preise für derartige Aufsatz-Klaviaturen liegen um die 100 Mark (einschließlich Software), was für weniger als zwei Oktaven auf Minitasten doch wohl etwas happig ist. Hier kann den Herstellern nur eine Preissenkung (etwa auf Joystickniveau, ohne Software) empfohlen werden. Denn für immerhin einen runden Hunderter wird sich so mancher Musik-Freak doch überlegen, ob er nicht weiter auf der Tastatur spielt. Überhaupt ist dem Anwender die Anschaffung eines solchen Auflage-Keyboards nur in dem Falle zu empfehlen, daß er sich mehr mit komponieren als mit spielen beschäftigt, denn zum Spielen sind zweieinhalb Oktaven doch etwas wenig.





Bis jetzt ging es nur um Zubehör, das zur Musikerzeugung den Soundchip des Computers durch geeignete Software ansteuert. Mehr und bessere Klänge als über die Tastatur des Computers kann man damit natürlich nicht erzeugen. Es existiert aber noch eine vollkommen andere Möglichkeit, den Commodore 64 als Hilfsmittel für professionelle Synthesizer einzusetzen: Durch Anschluß eines »richtigen« Synthesizers.

Professioneller Synthesizer am C 64

Mit einem Synthesizer der unteren Preisklasse lassen sich normalerweise in einer Speicherbank 32 Sounds ablegen. Zusätzliche ROMs (Speicherchips) kosten etwa 200 Mark. Was liegt da näher, als seinen Heimcomputer per MIDI-Interface — ein international genormtes Interface, für das es an den meisten Synthesizern einen Anschluß gibt — mit seinem Synthy zu verbinden, und so bis zu 320 Klangeinstellungen auf einmal im Computer zu speichern. Je nach Lust und Laune kann man seine interessantesten Kreationen dann auf Diskette bannen. Wenn Sie schon einmal auf einem komplexen Synthesizer gearbeitet haben, wissen Sie sicher, wieviel Mühe es kostet, einen gewünschten Sound zu erzeugen, da eine Vielzahl von Parametern eingestellt werden können. Eine für MIDI-Software bekannte Firma ist Jellinghouse. Sie vertreibt exzellente MIDI-Programme, die allerdings nicht ganz billig sind.

Ganz sicher ist aber der Anschluß eines Synthesizers per MIDI-Interface die professionellste Lösung für den Computer- **und** Musik-Freak, allerdings auch mit Abstand die kostspieligste.

Fazit

Wer sich nicht so stark für die Musikfähigkeiten seines C 64 interessiert, dem genügt sicherlich die Tastatur, um mit einem herkömmlichen Musikprogramm zu experimentieren, denn was soll ein Keyboard für einen halben Tausender, wenn es nach der anfänglichen Euphorie in der Ecke steht. Für solche Zwecke ist vielleicht eine Aufsatzklaviatur gar nicht ungeeignet, vorausgesetzt man bekommt sie zu einem günstigen Preis. Natürlich braucht man dazu ein passendes Programm, das die Tasten abfragt, die von der Klaviatur bedient werden.

Ein Keyboard kann man ohne Einschränkungen nur dann empfehlen, wenn Sie schon etwas Erfahrung im Umgang mit Klaviaturen haben, also zweihändig spielen können, oder zumindest den festen Vorsatz haben, es zu lernen. Außerdem sollten Sie natürlich echtes und dauerhaftes Interesse für elektronische Musikerzeugung haben.

Aber in diesem Falle ist es schon fast überlegenswert, ob die Anschaffung eines richtigen Synthesizers plus MIDI-Interface nicht sinnvoller ist. In Verbindung mit seinem Computer besitzt man dann ein System, das in musikalischer Hinsicht kaum noch Wünsche offen läßt.

Synthetische Musik ist ein anspruchsvolles Hobby für das man, wenn man wirklich begeistert davon ist, auch gerne etwas mehr investiert. Dabei ist ein richtiges Keyboard zusammen mit dem C 64 mit seinem doch schon beachtlich leistungsfähigen Sound-Chip natürlich ein guter Einstieg. Auf MIDI umsteigen kann man dann ja immer noch ...

(Michael Marek/ev)

Musik-Hardware zum C 64:
Colortone Keyboard, Waveform Co., 1912 Bonitaway, Berkeley CA 94704, USA
Microsound Keyboard, Autographics Limited,3a Reading Road, Henley on Thames, Oxon, RG91AB, Großbritannien
Ultisynth, Micro Händler, 4050 Mönchengladbach
Jellinghouse, Martineer Hellweg 40, 4600 Dortmund 70
Sequential Circuits, PO 16, 3640 AA Mijdrecht, Niederlande
Passport Design, Roland Späth, Colmarer Str. 16, 7000 Stuttgart 40
Wersiboard, Wersi Orgel- und Pianobausätze, Industriestr., 5401 Halsenbach

Klangprogrammierung ohne Ballast

Für viele C 64-Besitzer ist die Sound-Programmierung immer noch ein Buch mit sieben Siegeln. Wir zeigen hier, wie's funktioniert — ohne theoretischen Ballast.

Der im C 64 eingebaute Sound-Chip 6581 sorgt immer wieder für neue klangliche Überraschungen, wie viele Spiele und Synthesizerprogramme zeigen. Für sich allein ist dieser Chip allerdings noch kein Synthesizer, sondern er entfaltet seine klanglichen Fähigkeiten erst unter der Kontrolle eines Programms. Wenn man nun selbst mit Klangeffekten und Melodien experimentieren will, hat man prinzipiell zwei Möglichkeiten:

Fertige Programme

Man verwendet eines der mittlerweile sehr guten Synthesizer- oder Musikprogramme. Mit diesen Programmen kann man meistens alle für den Klang relevanten Werte (Parameter) komfortabel einstellen. Darüber hinaus kann man bei einigen Programmen ganze Musikstücke Note für Note eingeben und ähnlich wie bei einem Textprogramm editieren. Ein so erarbeitetes Werk kann man dann zwar samt Klang-Parametersatz auf Diskette abspeichern und wieder laden, man kann diese Melodien und Klänge normalerweise aber nicht in eigene Programme einbauen. In dem Kurs »Dem Klang auf der Spur«, die seit einiger Zeit im 64'er-Magazin erscheint, wird ein solches Programm Schritt für Schritt aufgebaut. Dieses Programm wird in seiner endgültigen Ausbaustufe den zuletzt genannten Nachteil nicht haben.

Selber programmieren

Wenn man in einem eigenen Programm nur einen Signalton (zum Beispiel einen Gong) oder einen anderen einfachen Klangeffekt benötigt, kann man ihn mit einigen POKE-Befehlen leicht selber programmieren. Dieser Artikel soll demjenigen, der sich nur am Rande mit der Klangprogrammierung beschäftigen will, zeigen, wie es geht.

Der erste Schritt

Man führe einmal folgende Befehlsfolge direkt am Bildschirm aus:

Lautstärke einstellen:	POKE 54296,15
Frequenz einstellen:	POKE 54272,0
	POKE 54273,40
Hüllkurve wählen:	POKE 64277,0
	POKE 54278,240
Ton an:	POKE 54276,33
Ton aus:	POKE 54276,32

Nach dem vorletzten Befehl sollte ein Dauerton hörbar werden, der nach dem letzten Befehl wieder verschwindet. Der SID (Sound Interface Device) wird über sogenannte Register programmiert. Diese Register belegen 29 Byte Speicherplatz im C 64 und zwar den Bereich 54272 bis 54300 (hexadezimal $D400 bis $D41C). Über den Inhalt dieser Register werden alle Funktionen des SID gesteuert (Tabelle 1). Der SID verfügt über drei voneinander unabhängige Stimmen, die jeweils durch sieben Register gesteuert werden. Wir befassen uns zunächst mit Stimme 1:

Frequenz (Register 54272 und 54273)

Diese beiden Register enthalten zusammen einen 16-Wert in der Folge Low-Byte, High-Byte. Man kann ihn wie folgt ermitteln:

F=PEEK(54272)+256*PEEK(54273)

Wenn man F durch 17,0284 dividiert, erhält man die tatsächliche Frequenz in Hz. Normalerweise möchte man aber Stimme 1 auf eine vorgegebene Frequenz programmieren, zum Beispiel 440 Hz. Den Wert F erhält man dann so:

F=440*17.0284

Diesen Wert zerlegt man in ein höher- und ein niederwertiges Byte (Low- und High-Byte) und POKEt diese in die beiden Frequenz-Register:

```
HI=INT(F/256)
LO=F-256*HI
POKE 54272,LO
POKE 54273,HI
```

Tabelle 2 enthält die Frequenzen der 12 Halbtöne einer Tonleiter zusammen mit den zugehörigen POKE-Werten. Um zu höheren oder tieferen Oktaven zu gelangen, muß man die Frequenzen lediglich verdoppeln (vervierfachen etc.) beziehungsweise halbieren (vierteln etc.). Die POKE-Werte müssen dann natürlich neu berechnet werden. Werden diese Frequenzen in einem Programm verwendet, so sollten die POKE-Werte aller benötigten Töne in zwei Feldern (einem für die Low- und einem für die High-Bytes) abgespeichert sein, damit das Programm keine Zeit mit der Berechnung dieser Werte verliert.

Das Steuerregister 54276

Über dieses Register wird der Ton ein- und ausgeschaltet. Man sollte vielleicht besser sagen: »angeschlagen« und »losgelassen«, da über die Hüllkurvensteuerung eine dynamische Beeinflussung der Lautstärke möglich ist. Darauf kommen wir später noch zurück. Im Steuerregister ist für das Ein- und Ausschalten Bit 0, das sogenannte GATE-Bit verantwortlich. Eine 1 im GATE-Bit schaltet die Stimme ein. Das Steuerregister legt auch die Kurvenform fest, die für das Klangbild der Stimme entscheidend ist. Eine Kurvenform wird ausgewählt, indem man eines der Bits 4 bis 7 setzt. Hier zunächst eine Tabelle mit den wichtigsten POKE-Werten für das Steuerregister:

Kurvenform	Bit	Ton an	Ton aus
Dreieck	4	17	16
Sägezahn	5	33	32
Rechteck	6	65	64
Rauschen	7	129	128

Die Werte sollte man gleich einmal ausprobieren. Um einen Rechteckklang zu erzeugen, sind allerdings noch weitere Maßnahmen notwendig (siehe unten). Das **Dreieck** klingt weich, dumpf und leiser als die anderen Kurvenformen. Es eignet sich für flötenähnliche Töne, aber auch für einen dezenten Signal-Gong. Der **Sägezahn** hat einen großen Obertongehalt und klingt damit sehr hell. Sägezahn ist die Standard-Kurvenform für die meisten musikalischen Anwendungen. Um eine **Rechteck**-Kurve zum Klingen zu bringen, muß man im SID zusätzlich die sogenannte **Pulsweite** einstellen. Dieser Wert steuert das Zeitverhältnis der beiden Pegel, zwischen denen die Rechteck-Kurve hin- und herspringt. Die Register 54274 und 54275 (Low- und High-Byte) sind für die Pulsweite maßgeblich. Es werden allerdings nur 12 Bit berücksichtigt, was einem Bereich von 0 bis 4095 entspricht. Der Wert 2048 (Bereichsmitte) entspricht dabei einer

Adresse	Reg.	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	Registername	
54272	0	F7	F6	F5	F4	F3	F2	F1	F0	Frequenz low	Schreib-Register, Stimme 1
54273	1	F15	F14	F13	F12	F11	F10	F9	F8	Frequenz high	
54274	2	P7	P6	P5	P4	P3	P2	P1	P0	Pulsweite low	
54275	3	unbenutzt	unbenutzt	unbenutzt	unbenutzt	P11	P10	P9	P8	Pulsweite high	
54276	4	Rauschen	Rechteck	Sägezahn	Dreieck	Test	Ringmodulat.	Synchronisat.	GATE	Kontrollregister	
54277	5	Attack 3	Attack 2	Attack 1	Attack 0	Decay 3	Decay 2	Decay 1	Decay 0	Attack/Decay	
54278	6	Sustain 3	Sustain 2	Sustain 1	Sustain 0	Release 3	Release 2	Release 1	Release 0	Sustain/Release	
54279	7	F7	F6	F5	F4	F3	F2	F1	F0	Frequenz low	Schreib-Register, Stimme 2
54280	8	F15	F14	F13	F12	F11	F10	F9	F8	Frequenz high	
54281	9	P7	P6	P5	P4	P3	P2	P1	P0	Pulsweite low	
54282	10	unbenutzt	unbenutzt	unbenutzt	unbenutzt	P11	P10	P9	P8	Pulsweite high	
54283	11	Rauschen	Rechteck	Sägezahn	Dreieck	Test	Ringmodulat.	Synchronisat.	GATE	Kontrollregister	
54284	12	Attack 3	Attack 2	Attack 1	Attack 0	Decay 3	Decay 2	Decay 1	Decay 0	Attack/Decay	
54285	13	Sustain 3	Sustain 2	Sustain 1	Sustain 0	Release 3	Release 2	Release 1	Release 0	Sustain/Release	
54286	14	F7	F6	F5	F4	F3	F2	F1	F0	Frequenz low	Schreib-Register, Stimme 3
54287	15	F15	F14	F13	F12	F11	F10	F9	F8	Frequenz high	
54288	16	P7	P6	P5	P4	P3	P2	P1	P0	Pulsweite low	
54289	17	unbenutzt	unbenutzt	unbenutzt	unbenutzt	P11	P10	P9	P8	Pulsweite high	
54290	18	Rauschen	Rechteck	Sägezahn	Dreieck	Test	Ringmodulat.	Synchronisat.	GATE	Kontrollregister	
54291	19	Attack 3	Attack 2	Attack 1	Attack 0	Decay 3	Decay 2	Decay 1	Decay 0	Attack /Decay	
54292	20	Sustain 3	Sustain 2	Sustain 1	Sustain 0	Release 3	Release 2	Release 1	Release 0	Sustain/Release	
54293	21	unbenutzt	unbenutzt	unbenutzt	unbenutzt	unbenutzt	GF 2	GF 1	GF 0	Grenzfrequenz low	Schreib-Register, Filter
54294	22	GF 10	GF 9	GF 8	GF 7	GF 6	GF 5	GF 4	GF 3	Grenzfrequenz high	
54295	23	Resonanz 3	Resonanz 2	Resonanz 1	Resonanz 0	Filtex	Filter 3	Filter 2	Filter 1	Resonanz/Filter	
54296	24	Aus	Hochpass	Bandpass	Tiefpass	L 3	L 2	L 1	L 0	Mode/Lautstärke	
54297	25	Pot X 7	Pot X 6	Pot X 5	Pot X 4	Pot X 3	Pot X 2	Pot X 1	Pot X 0	Potentiometer X	Lese-Register
54298	26	Pot Y 7	Pot Y 6	Pot Y 5	Pot Y 4	Pot Y 3	Pot Y 2	Pot Y 1	Pot Y 0	Potentiometer Y	
54299	27	O7	O6	O5	O4	O3	O2	O1	O0	Oszillator 3	
54300	28	H7	H6	H5	H4	H3	H2	H1	H0	Hüllkurve Osz. 3	

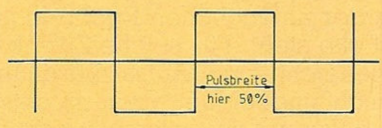


Tabelle 1. Übersicht über die SID-Register

Note	Frequenz	SID-Low-Byte	SID-High-Byte
c^1	261,63	103	17
cis^1	277,18	112	18
d^1	293,66	137	19
dis^1	311,13	178	20
e^1	329,63	237	21
f^1	349,23	59	23
fis^1	369,99	156	24
g^1	329,00	19	26
gis^1	415,30	160	27
a^1	440,00	68	29
ais^1	466,16	2	31
h^1	493,88	218	32

Dieses kleine Programm liefert die Tabelle:

```
10 FOR I=0 TO 11
20 : F=440*(2↑((I−9)/12))
30 : FS=INT(F*17.0284+.5)
40 : HI=INT(FS/256)
50 : LO=FS−256*HI
60 : PRINT F,LO,HI
70 : NEXT I
```

Tabelle 2. Frequenz der 12 Halbtöne einer Tonleiter (»eingestrichene Oktave«)

symmetrischen Rechteckkurve, das heißt einer mit einem Zeitverhältnis 1:1. Diese Einstellung erreicht man durch:

POKE 54274,0 :REM LOW-BYTE
POKE 54275,8 :REM HIGH-BYTE

Jetzt sollte nach:

POKE 54276,65

ein Rechteckton hörbar sein. Das Rechteck klingt im Vergleich zum Sägezahn hohl und erinnert daher ein wenig an eine Klarinette. Über die Pulsweitensteuerung ist das Klangbild des Rechtecks allerdings sehr variabel.

Klangvielfalt durch Hüllkurven

Töne natürlicher Instrumente haben selten einen konstanten Lautstärkeverlauf, wenn man einmal von der Orgel absieht. Töne von Blasinstrumenten brauchen eine gewisse Zeit, bis sie vom Ansatz zur vollen Lautstärke anschwellen. Ein angeschlagener Klavierton setzt zwar unmittelbar ein, aber klingt dann beim Halten der Taste erst schnell und dann langsamer aus. Auch beim Loslassen der Taste reißt der Ton nicht schlagartig ab, sondern verklingt innerhalb einiger Sekundenbruchteile. Diese Dynamik kann man auch mit dem SID erzeugen. Sobald das GATE-Bit im Kontrollregister gesetzt ist, folgt die Lautstärke der Stimme einer Kurve, deren Verlauf durch die Register 54277 und 54278 gesteuert wird. In dem Beispiel »Der erste Schritt« wurde eine einfache rechteckförmige Hüllkurve eingestellt (Orgelcharakter). Vor dem systematischen Teil zunächst einige Experimente:

Man führe die POKEs aus »Der erste Schritt« aus, damit die Befehle alle zusammen auf dem Bildschirm stehen. Nun kann man die POKEs für die Hüllkurve ändern und sich den Effekt über »Ton an« und »Ton aus« anhören:

Längeres Ausklingen:
POKE 54278,250

Klavierartiger Anschlag:
POKE 54277,6
POKE 54278,250

Anschwellen wie bei Bläsern:
POKE 54277,96
POKE 54278,247

A D S R — Hüllkurven

Die SID-Hüllkurve (Bild 1) gliedert sich zeitlich in vier Phasen. Jede Phase kann dabei unabhängig in 16 Stufen gesteuert werden.

Attack

Beim Setzen des GATE-Bits schwillt der Ton von Null bis zur maximalen Lautstärke an. Die Zeit, in der das geschieht, steuert der A-Parameter. Diese Zeit reicht von 2 ms (A = 0, unhörbar kurz) bis 8 s (A = 15).

Decay

Nach Erreichen des Maximalpegels fällt die Lautstärke wieder ab, sofern ein Haltepegel (Sustain) kleiner als 15 eingestellt ist. Die Decay-Zeit wird über den D-Parameter gesteuert. Sie reicht von 6 ms (D = 0) bis 24 s (D = 15).

Sustain

Nach der Decay-Phase bleibt der Ton auf einem konstanten Haltepegel (Sustain) stehen. Im Falle S = 15 (Maximalpegel) kann von einer Decay-Phase nicht gesprochen werden. In diesem Fall steigt die Lautstärke in der Attack-Phase auf Maximalpegel um dort zu bleiben. Im Falle S = 0 klingt der Ton in der Decay-Phase vollständig aus. (das heißt bis auf Null).

Release

Die Sustain-Phase dauert so lange, wie das GATE-Bit gesetzt ist. (Auf diese Weise kann man einen Dauerton erzeugen). Durch das Rücksetzen des GATE-Bit wird die Release-Phase eingeleitet. Der Ton klingt vom momentanen Pegel (falls dieser nicht schon Null ist) auf Null aus. Diese Ausklingzeit ist durch den R-Parameter im Bereich 6 ms (R = 0) bis 24 s (R = 15) steuerbar. Die Release-Phase kann übrigens jederzeit durch Zurücksetzen des GATE-Bits eingeleitet werden, auch wenn sich die Hüllkurve noch in der Attack-, oder der Decay-Phase befindet. Der Ton klingt dann sofort vom momentanen Pegel in der eingestellten Zeit auf Null ab. Man spricht dann auch von einer ADR- oder einer AR-Hüllkurve.

Die Hüllkurvenregister 54277 und 54278

Jedes dieser Register muß man sich in zwei 4-Bit-Hälften aufgeteilt denken, von denen jede einen der Parameter enthält:

Parameter	Bits	Register
A	7-4	54277
D	3-0	54277
S	7-4	54278
R	3-0	54278

Die Parameter A und D beziehungsweise S und R können immer nur zusammen in ihre Register geschrieben werden. Die Parameterwerte auf den Bitpositionen 3-0 kann man dabei direkt übernehmen,

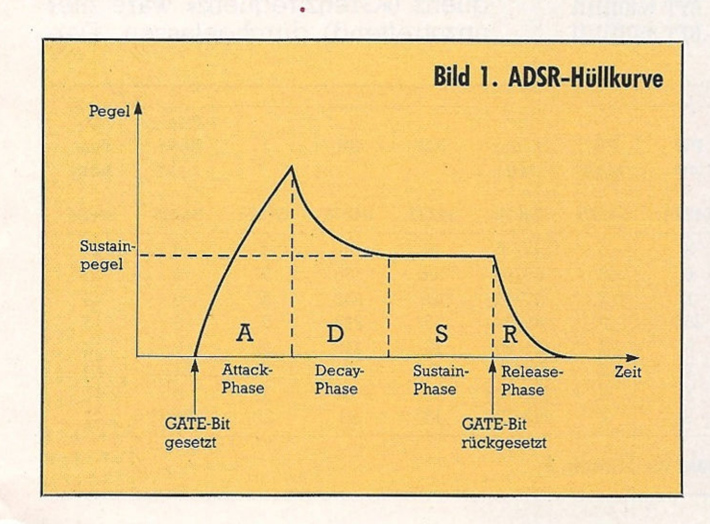


Bild 1. ADSR-Hüllkurve

Tabelle 3. Hüllkurven-Beispiele

A	D	S	R	POKE 54277	POKE 54278	Form
0	0	15	0	0	240	
0	0	15	10	0	250	
0	9	4	10	9	74	
6	0	15	9	96	249	
6	0	15	0	96	240	
8	0	0	0	128	0	
6	9	4	10	105	74	
0	9	0	9	9	9	
13	14	0	12	222	12	

während man die Werte auf den Positionen 7-4 um vier Bitpositionen nach links schreiben muß, was man in Basic durch Multiplikation mit 16 erreicht:
POKE 54277, 16*A+D
POKE 54278, 16*S+R

Tabelle 3 zeigt einige Beispiel-Hüllkurven mit den dazugehörenden POKE-Werten. Auch diese Hüllkurven sollte man einmal ausprobieren. Damit sind die wichtigsten Register, die für die Erzeugung einfacher Klänge benötigt werden, bekannt. Die Steuerung der Stimmen 2 und 3 erfolgt analog zu Stimme 1 über je sieben Register.

Stimme	1	2	3
Frequenz	54272 54273	54279 54280	54286 54287
Pulsweite	54274 54275	54281 54282	54288 54289
Steuer-register	54276	54283	54290
AD	54277	54284	54291
SR	54278	54285	54292

Spezialeffekte

Wir wenden uns nun den Spezialeffekten zu. Dazu gehören Synchronisation, Ringmodulation und Filterung. Wir betreten damit bereits das Gebiet der fortgeschrittenen Sound-Programmierung. Notgedrungen wird es auch etwas komplizierter.

Synchronisation

Wenn man Bit 1 des Steuerregisters von Stimme 1 setzt, wird das Ausgangssignal von Stimme 1 durch das Signal von Stimme 3 synchronisiert, **das heißt** phasenstarr gekoppelt. Resultat ist ein mit Worten schwer zu beschreibender, in hohem Maße »elektronischer« Klang. Auch die Stimmen 2 und 3 lassen sich durch Setzen von Bit 1 der jeweiligen Steuerregister synchronisieren. Dabei wird Stimme 2 von Stimme 1 synchronisiert und synchronisiert ihrerseits Stimme 3.

Ringmodulation

Durch Setzen von Bit 2 des Steuerregisters von Stimme 1 entsteht ein Ringmodulator-Produkt der Dreieckskurven der Stimmen 1 und 3. Hörbar wird es nur dann, wenn Stimme 1 zusätzlich auf Dreieck eingestellt ist (Bit 4 gesetzt). Bei der Ringmodulation werden zwei Signale multiplikativ überlagert. Das Resultat sind meistens metallische Klänge, die sich für Glocken und Gongs eignen. Auch die Ringmodulation ist in jeder Stimme möglich. Wie bei der Synchronisation modulieren sich die Stimmen nach dem Schema:

1 → 2 → 3 → 1

Mehrere Kurvenformen gleichzeitig

Weitere Kurvenformen kann man noch dadurch erhalten, in dem man von der Kurvenform Bit 4 bis 7 mehrere zugleich setzt. Dabei werden allerdings nicht mehrere Formen zugleich hörbar, sondern es entstehen neue Kurvenformen. Erfahrungsgemäß bringen nur die Kombinationen:
Rechteck — Sägezahn (Bit 6 und 5)
sowie:
Rechteck — Dreieck (Bit 6 und 4)
brauchbare Resultate. Bei diesen Kombinationen hat auch die Pulsweiteneinstellung einen Einfluß auf den Klang.

Setzen und Rücksetzen einzelner Bits

Bei der Programmierung des Steuerregisters will man oft einzelne Bits setzen oder zurücksetzen, ohne daß man dabei die anderen Bits verändern möchte. Diese Aufgabe erledigt man am besten mit Hilfe zweier Sätze von jeweils acht sogenannten »Masken«. Der erste Satz besteht aus Bytes, bei denen immer genau ein Bit gesetzt ist, der zweite besteht aus Bytes, bei denen immer genau ein Byte 0 ist. Diese 16 Masken kann man sich wie folgt definieren:

Basic	Assembler
DIM	ON .BYT %00000001
ON%(7),OF%(7)	.BYT %00000010
FOR I=0 TO 7:	
ON%(I)=2↑I:	.BYT %00000100
OF%(I)=255-2 ↑I	.BYT %00001000
NEXT I	.BYT %00010000
	.BYT %00100000
	.BYT %01000000
	.BYT %10000000
	OF .BYT %1111110
	.BYT %1111101
	.BYT %1111011
	.BYT %1110111
	.BYT %1101111
	.BYT %1011111
	.BYT %0111111

Nun kann man Bit N im Steuerregister sehr einfach setzen:

Basic	Assembler
SR=54276	LDX N
POKE SR,	LDA $D404
PEEK(SR) OR	ORA ON,X
ON%(N)	STA $D404

Das Löschen von Bit N geschieht ganz ähnlich:

Basic	Assembler
SR=54276	LDX N
POKE SR,	LDA $D404
PEEK(SR) AND	AND OF,X
OF%(N)	STA $D404

Filterung

Das im SID vorhandene Filter ist vergleichbar mit dem Klangregler einer Audio-Anlage. Gegenüber dem bekannten Einstellsystem für Höhen und Bässe bestehen aber auch Unterschiede. Beim SID-Filter ist eine Grenzfrequenz einstellbar, oberhalb oder unterhalb der eine Absenkung der Frequenzanteile des zu filternden Signals erfolgen soll. Das Filter kennt drei Betriebsarten:

Tiefpaß

Nur Frequenzanteile unterhalb der Grenzfrequenz werden durchgelassen. Anteile oberhalb der Grenzfrequenz werden mit zunehmendem Abstand von dieser zunehmend unterdrückt.

Hochpaß

Frequenzanteile oberhalb der Grenzfrequenz werden durchgelassen. Anteile unterhalb der Grenzfrequenz werden unterdrückt. Die Hochpaß-Betriebsart erzeugt beim SID meistens einen leisen dünnen Klang.

Bandpaß

Es werden nur die Frequenzanteile in der Umgebung der Mittenfrequenz (»Grenzfrequenz« wäre hier unzutreffend) durchgelassen. Dar-

	F high	PW low	PW high	Steuer-reg.	AD	SR	F	Filter Reso-nanz	Mo-dus/ Laut
	54273	54274	54275	54276	54277	54278	54294	54295	54296
Signalton	60	X	X	17/16	10	10	X	0	15
Klarinette	20	0	8	65/64	106	135	X	0	15
Flöte	40	X	X	17/16	128	248	X	0	15
Oboe/Fagott	30/10	250	0	65/64	73	248	X	0	15
Schnarre	5	0	8	81/80	73	248	X	0	15
Banjo	30	X	X	32/32	8	8	50	241	111
Schuß	200	X	X	129/128	9	9	X	0	15
Explosion	1	X	X	129/128	13	93	20	241	31

Tabelle 4. Einstellbeispiele für Stimme 1

über hinaus kann man die Filterbetriebsarten auch kombinieren. Ein Beispiel dafür ist die

Bandsperre

Diesen Modus erhält man, wenn man die Betriebsarten Hochpaß und Tiefpaß zugleich anwählt. Es werden Frequenzen in der Umgebung der Mittenfrequenz abgeschwächt. Eingestellt werden die Filterbetriebsarten durch die Bits 6, 5 und 4 des Registers 54296, dem gleichen Register, mit dem auch die Gesamtlautstärke des SID gesteuert wird (Bits 3 bis 0)

Bit 6 = 1 Hochpaß
Bit 5 = 1 Bandpaß
Bit 4 = 1 Tiefpaß

Über die Bits 2 bis 0 des Registers 54295 kann man steuern, welche der einzelnen Stimmen gefiltert werden sollen. Eine 1 bedeutet dabei Filterung.

Bit 0 entspricht Stimme 1
Bit 1 entspricht Stimme 2
Bit 2 entspricht Stimme 3

Eine 0 besagt, daß die Stimme ungefiltert an den Ausgang gelangen soll.

Die Bits 4 bis 7 des gleichen Registers 54295 enthalten einen 4-Bit-Wert, über den man die Resonanz des Filters (in der Elektrotechnik auch »Güte« genannt) steuern kann. Eine hohe Resonanz (der Maximalwert ist 15) bewirkt eine zusätzliche Verstärkung der Frequenzanteile in der Nähe der Grenz- beziehungsweise Mittenfrequenz. Diese Verstärkung kann beim SID auch zu klanglich interessanten Verzerrungen führen (E-Gitarre). Unerwähnt ist bis hierher die Einstellung der Grenz- beziehungsweise Mittenfrequenz geblieben. Sie wird über das Register 54294 eingestellt (volle 8 Bit). Es besteht zudem die Möglichkeit einer Feineinstellung über die Bits 0-2 des Registers 54293. Eine solche Feineinstellung bleibt aber erfahrungsgemäß unhörbar. Man kann also Register 54293 getrost unberücksichtigt lassen.

Eine kleine Anleitung zum Experimentieren mit dem Filter

Man stelle Stimme 1 auf Dauerton ein (zum Beispiel mit der POKE-Folge ganz am Anfang dieses Artikels). Man fahre so fort:
Tiefpaßeinstellung, Lautstärke=15:
POKE 54296,31
Resonanz=15, Stimme 1 über das Filter schicken: POKE 54295,241

Jetzt ist vermutlich nichts mehr oder kaum noch etwas zu hören, weil die Grenzfrequenz mit 0 voreingestellt ist. Mit der Grenzfrequenz kann man jetzt experimentieren:
POKE 54294,20 (40,60,80...)

Das Klangbild von Stimme 1 müßte jetzt zunehmend heller werden. Zum Abschluß folgen in Tabelle 4 noch einige Einstellbeispiele für Stimme 1. Die Einstellungen, bei denen das Filter benutzt wird, sind mit Vorsicht zu genießen, da Filtereinstellungen schlecht reproduzierbar sind. In diesem Punkt weisen die einzelnen SID-Bausteine beachtliche Streuungen auf.

Wer sich weitergehend mit der Klang- und Musikprogrammierung beschäftigen will, der sei auf die schon erwähnte Reihe »Dem Klang auf der Spur« verwiesen. In Ausgabe 2/85 des 64'er-Magazins findet man mehrere kleine Beispielprogramme zur SID-Programmierung. In Ausgabe 7/85 wird ein größeres Programm beschrieben, mit dem man unter anderem alle Parameter des SID interaktiv und komfortabel beeinflussen kann. In Ausgabe 10/85 wird ein Sequenzer-Programm erscheinen, das dreistimmige Melodien zeitexakt spielen kann, während nebenbei auch noch ein anderes Programm läuft.

(Thomas Krätzig/ev)

Sound Machine

Mit Sound Machine können nun auch Sie, wie in professionellen Spielen, unabhängige Hintergrundmusik programmieren.

Sound Machine besteht aus zwei Programmen, einem Maschinenprogramm, das die Musik später spielt, und einem Basic-Programm, mit dem Sie die Melodie einfach eingeben können.

Das Maschinenprogramm Sound Machine (Listing 1) belegt den Speicherplatz von dezimal 53004 bis 53187. Nach dem Abtippen mit dem MSE kann es mit »LOAD "SOUND MACHINE",8,1« jederzeit geladen werden. Es benutzt nur die erste Stimme des SID, die anderen bleiben für Geräuscheffekte frei. Eine Melodie wird mit SYS 53004 gestartet.

Die Noten müssen ab 49152 im Speicher stehen. Zur Noteneingabe können Sie Listing 2 verwenden.

Die Melodie kann mit SYS 53027 gestoppt werden.

Bevor man das Programm Noteneingabe startet, muß sich Sound Machine schon im Speicher befinden.

```
 programm : sound machine   cf0c cfc3
----------------------------------------

cf0c : a9 00 8d 00 d4 8d 01 d4  80
cf14 : 8d 07 dc 20 31 cf 20 59  25
cf1c : cf a9 59 8d 0f dc 60 a9  75
cf24 : 08 8d 0f dc a9 00 8d 04  2b
cf2c : d4 8d 18 d4 60 a9 20 8d  56
cf34 : 02 d4 a9 08 8d 03 d4 a9  a3
cf3c : 18 8d 05 d4 a9 52 8d 06  66
cf44 : d4 a9 21 8d 04 d4 a9 0f  93
cf4c : 8d 18 d4 a2 00 a0 c0 86  84
cf54 : a4 84 a5 58 60 78 a9 67  ee
cf5c : 8d 14 03 a9 cf 8d 15 03  ad
cf64 : 58 60 ea 78 ad 0d dc 29  bf
cf6c : 02 f0 03 20 77 cf 58 4c  9b
cf74 : 31 ea ea a9 00 8d 07 dc  4c
cf7c : a9 18 8d 0f dc a9 10 8d  ed
cf84 : 04 d4 a0 00 b1 a4 8d 00  91
cf8c : d4 c8 b1 a4 8d 01 d4 f0  db
cf94 : 05 a9 41 8d 04 d4 c8 b1  dd
cf9c : a4 8d 06 dc f0 14 ea 18  af
cfa4 : a5 a4 69 03 85 a4 90 02  1a
cfac : e6 a5 a9 59 8d 0f dc ea  95
cfb4 : 60 ea a9 00 85 a4 a9 c0  99
cfbc : 85 a5 a9 59 8d 0f dc 60  2f
```

Listing 1. Das Maschinenprogramm »Sound Machine« muß mit dem MSE eingegeben werden.

Das Programm Noteneingabe löscht als erstes den Notenspeicher. Es fragt dann nach der Zeit, die eine Viertelnote gespielt werden soll. Durch diesen Wert wird also die Geschwindigkeit ihrer Melodie festgelegt.

Jetzt kann die eigentliche Noteneingabe beginnen:

Auf die Frage »NOTE : ?« gibt man zuerst die Notenbezeichnung, dann die Oktave ein. Einige Beispiele wären: »C5«, »A#3«. Erniedrigte Noten können nicht direkt eingegeben werden. Hb3 muß also als »A#3« eingegeben werden. Pausen gibt man mit »P« oder »PAUSE« ein.

Als Antwort auf die Frage »ZEIT : ?« gibt man die Notenlänge im Format x/y ein. Zwei Beispiele: Eine ganze Note wird als »1/1«, eine punktierte

Um die Eingabe von Musikstücken zu vereinfachen, versteht das Programm folgende Befehle:

Befehl	Bedeutung
—	Notenzähler um eins verringern. Die Note wird nicht gelöscht, kann aber überschrieben werden
+	Notenzähler um eins erhöhen
SAVE	Abspeichern der Noten. Es werden soviel Noten abgespeichert, wie der Notenzähler angibt. Dahinter werden drei Nullen (für Notenende) geschrieben. So abgespeicherte Notenfiles können im Direktmodus mit »LOAD 'Filename' (,8,1)« geladen werden
LOAD	Notenfile laden oder (nur bei Diskette) das Directory anzeigen
DEF	Eine Notenreihe definieren. Es können maximal sechs Reihen à 200 Noten definiert werden
INSERT	Eine definierte Notenreihe anhängen
COPY	Kopieren mehrerer aufeinanderfolgender Noten. Gibt man die erste Note größer an als die letzte, so werden die Noten in umgekehrter Reihenfolge kopiert
ON	Sound Machine starten (SYS 53004)
OFF	Sound Machine stoppen (SYS 53027)
QUIT	Programm beenden
NEW	Setzt den Notenzähler auf die erste Note. Die Noten werden nicht (!) gelöscht
SET END	Musikende setzen
CLEAR	Völliger Neustart des Programmes, alle Noten werden gelöscht

halbe als »3/4« eingegeben. x kann maximal doppelt so groß wie y sein, für y sind die Werte 1, 2, 4, 8, 16, 32 und 64 erlaubt.

Aus jeder der gewählten Funktionen kommt man mit »↑« zurück ins Hauptprogramm. Das Programm geht davon aus, daß ein Floppy-Laufwerk 1541 am seriellen Bus angeschlossen ist. Beabsichtigt man, das Programm mit einer Datasette zu betreiben, so sind einige Zeilen wie folgt zu ändern:

```
1130  OPEN 1,1,1,A$
1185  CLOSE 1
1190  PRINT:PRINT »TASTE
      DRUECKEN«
2050  REM
2055  OPEN 1,1,0,A$
2180  REM
2185  PRINT:PRINT »TASTE
      DRUECKEN«
```

(Arno Muhr/bs)

```
100 REM ******************************    <238>
110 REM * + -     NOTENEINGABE     - + *  <024>
120 REM * ZUM PROGRAMM 'SOUND-MACHINE'*   <211>
130 REM * FUER DEN C64                *   <045>
140 REM * VON:    ARNO MUHR           *   <148>
150 REM *         UNTERSTR. 32        *   <190>
160 REM *         5176 INDEN-FRENZ    *   <142>
170 REM ******************************    <052>
180 POKE 53280,9:POKE 53281,5:PRINT"{BLACK
    }"                                    <061>
190 PRINT"{CLR,CTRL-H,DOWN,CTRL-N,RIGHT}BI
    TTE CA. 13{2SPACE}SEKUNDEN WARTEN"    <022>
200 NA=49152:FOR I=NA TO NA+3000:POKE I,0:
    NEXT                                  <028>
210 DIM NW(5,203),T(5,203):GOSUB 320      <034>
220 INPUT"{CLR,DOWN,RIGHT}ZEIT FUER 1/4 NO
    TE (1-120):";GT                       <243>
230 GT=INT(GT):GT=GT/120*32:IF GT<=0 OR GT
    >32 THEN 220                          <065>
240 PRINT CHR$(147)                       <013>
250 PRINT CHR$(19);C/3+1"{LEFT,SPACE}. NOT
    E : ":N$=""                           <218>
260 INPUT "NOTE{2SPACE}: ";N$             <207>
270 GOTO 610                              <000>
280 W=1                                   <031>
290 FOR I=1 TO 98:IF N$=N$(I) THEN W=W(I) <253>
300 NEXT                                  <056>
310 RETURN                                <114>
320 REM *** NOTENWERTE EINLESEN ***       <230>
330 DIM N$(98),W(98)                      <140>
340 FOR I=1 TO 98:READ N$                 <058>
350 N$(I)=N$:NEXT                         <154>
360 FOR I=1 TO 98:READ W                  <084>
370 W(I)=W:NEXT                           <097>
380 RETURN                                <184>
390 REM *** NOTENDATAS ***                <046>
400 DATA C0,C#0,D0,D#0,E0,F0,F#0,G0,G#0,A0
    ,A#0,H0                               <001>
410 DATA C1,C#1,D1,D#1,E1,F1,F#1,G1,G#1,A1
    ,A#1,H1                               <103>
420 DATA C2,C#2,D2,D#2,E2,F2,F#2,G2,G#2,A2
    ,A#2,H2                               <207>
430 DATA C3,C#3,D3,D#3,E3,F3,F#3,G3,G#3,A3
    ,A#3,H3                               <053>
440 DATA C4,C#4,D4,D#4,E4,F4,F#4,G4,G#4,A4
    ,A#4,H4                               <157>
450 DATA C5,C#5,D5,D#5,E5,F5,F#5,G5,G#5,A5
    ,A#5,H5                               <003>
460 DATA C6,C#6,D6,D#6,E6,F6,F#6,G6,G#6,A6
    ,A#6,H6                               <107>
470 DATA C7,C#7,D7,D#7,E7,F7,F#7,G7,G#7,A7
    ,A#7,H7                               <209>
480 DATA PAUSE,P                          <114>
490 DATA 268,284,301,318,337,358,379,401,4
    25,451,477,506,536,568,602,637,675    <254>
500 DATA 716,758,803,851,902,955,1012,1072
    ,1136,1204,1275,1351,1432,1517,1607   <037>
510 DATA 1703,1804,1911,2025,2145,2273,240
    8,2551,2703,2864,3034,3215,3406,3608  <124>
520 DATA 3823,4050,4291,4547,4817,5103,540
    7,5728,6069,6430,6812,7217,7647,8101  <202>
530 DATA 8583,9094,9634,10207,10814,11457,
    12139,12860,13625,14435,15294,16203   <031>
540 DATA 17167,18188,19269,20415,21629,229
    15,24278,25721,27251,28871,30588      <036>
550 DATA 32407,34334,36376,38539,40830,432
    58,45830,48556,51443,54502,57743      <236>
560 DATA 61176,64814,0,0                  <133>
570 :                                     <038>
580 REM   ******************              <115>
590 REM   *+- FUNKTIONEN -+*              <082>
600 REM   ******************              <135>
610 :                                     <078>
620 IF N$="-" AND C=>3 THEN C=C-3:GOTO 250 <157>
630 IF N$="-"        THEN PRINT "ERSTE NOTE
    !{2SPACE}":GOTO 250                   <115>
640 IF N$="+"        THEN C=C+3:GOTO 250  <160>
650 IF N$="SAVE"     THEN 1120            <063>
660 IF N$="LOAD"     THEN 1870            <234>
670 IF N$="DEF."     THEN 1280            <150>
680 IF N$="COPY"     THEN 1650            <105>
690 IF N$="INSERT"   THEN 1470            <033>
700 IF N$="QUIT"     THEN 1040            <103>
710 IF N$="ON"       THEN SYS 53004:GOTO 250 <234>
720 IF N$="OFF"      THEN SYS 53027:GOTO 250 <238>
730 IF N$="NEW"      THEN C=0:GOTO 240    <110>
740 IF N$="CLEAR"    THEN RUN             <110>
750 IF N$="SET END"THEN 1060              <052>
760 GOSUB 280                             <076>
770 INPUT "ZEIT{2SPACE}: ";T$:T$=T$+"{2SPA
    CE}"                                  <089>
780 PRINT "{26SPACE,26LEFT}";             <043>
790 IF W=1 THEN PRINT "NOTEN FEHLER!":IF Z
    Z<>1 GOTO 250                         <042>
800 IF W=1 THEN RETURN                    <182>
810 HB=INT(W/256):LB=W-256*HB             <229>
820 A=VAL(T$):AA=3:IF A>9 THEN AA=4       <179>
830 A$=MID$(T$,AA,2):T=0                  <149>
840 A1=VAL(A$)                            <223>
850 IF A1=64 THEN T=2                     <074>
860 IF A1=32 THEN T=4                     <162>
870 IF A1=16 THEN T=8                     <078>
880 IF A1=08 THEN T=16                    <068>
890 IF A1=04 THEN T=32                    <028>
900 IF A1=02 THEN T=64                    <093>
910 IF A1=01 THEN T=127                   <132>
920 IF T=0 THEN PRINT" ZEIT FEHLER!":IF ZZ
```

Listing 2. Mit diesem Programm können einfach Musikstücke programmiert werden.

```
    <>1 THEN GOTO 250                         <188>
930 IF T=0 THEN RETURN                        <217>
940 T=T/32*GT:T=T*A                           <026>
950 IF T>255 THEN PRINT " ZEIT ZU LANG !":
    IF ZZ<>1 GOTO 250                         <254>
960 IF T>255 THEN RETURN                      <162>
970 IF ZZ=1 THEN RETURN                       <120>
980 POKE 49152+C,LB                           <094>
990 POKE 49153+C,HB                           <200>
1000 POKE 49154+C,T                           <143>
1010 C=C+3                                    <239>
1020 GOTO 250                                 <018>
1030 :                                        <246>
1040 FOR I=0 TO 9:POKE 49152+C+I,0:NEXT       <002>
1050 END                                      <036>
1060 REM *** SET END ***                      <194>
1070 REM -ENDE DES STUECKES MAKIEREN          <085>
1080 POKE NA+C,0                              <143>
1090 POKE NA+C+1,0                            <241>
1100 POKE NA+C+2,0                            <252>
1110 GOTO 250                                 <110>
1120 REM *** SAVE ***                         <205>
1130 REM -ABSPEICHERN VON MUSIKSTUECKEN       <178>
1140 SYS 53 TAN 27                            <017>
1150 I=0:PRINT "{CLR,DOWN,RIGHT}ABSPEICHER
     N "                                      <008>
1160 INPUT "{CLR,DOWN,RIGHT}TITEL ";A$        <214>
1170 IF LEN(A$)>16 THEN 1150                  <243>
1180 IF A$="↑" THEN 240                       <008>
1190 OPEN 1,8,1,A$                            <095>
1200 PRINT#1,CHR$(0);CHR$(192);               <019>
1210 :NP=PEEK(49152+I)                        <172>
1220 :PRINT#1,CHR$(NP);                       <128>
1230 :I=I+1:IF I<=C THEN 1210                 <025>
1240 PRINT#1,CHR$(0);CHR$(0);CHR$(0)          <171>
1250 CLOSE 1:GOSUB 2380                       <006>
1260 PRINT"{DOWN,RIGHT}";E;Z$:PRINT"{DOWN,
     RIGHT}TASTE DRUECKEN "                   <229>
1270 POKE 198,0:WAIT 198,1:POKE 198,0:GOTO
     240                                      <017>
1280 REM *** DEF. ***                         <112>
1290 REM -DEFINIEREN EINER NOTENFOLGE         <161>
1300 PRINT "{CLR}":NN=0                       <058>
1310 IF NS>5 THEN PRINT"ES BESTEHEN SCHON
     6 ":FOR I=0 TO 999:NEXT:PRINT"{CLR}":
     GOTO 250                                 <104>
1320 PRINT"{HOME,DOWN,RIGHT}DEFINITIONFILE
     NR.";NS:PRINT NN+1;"{LEFT,SPACE}. NO
     TE"                                      <079>
1330 INPUT "{DOWN}NOTE{2SPACE}: ";N$          <188>
1340 IF N$="END" THEN 1430                    <192>
1350 IF N$="↑" THEN 240                       <028>
1360 IF N$="-" THEN NN=NN-1:GOTO 1320         <001>
1370 IF N$="+" THEN NN=NN+1:GOTO 1320         <138>
1380 GOSUB 280:ZZ=1:GOSUB 770:ZZ=0            <095>
1390 IF T=0 OR T>255 OR W=1 THEN 1320         <003>
1400 NW(NS,NN)=W:T(NS,NN)=T                   <167>
1410 NN=NN+1:IF NN<201 THEN 1320              <092>
1420 PRINT "DEFINITIONFILE IS VOLL!"          <028>
1430 W=0:T=0:FOR I=1 TO 2:NN=NN+1:NW(NS,NN
     )=0:T(NS,NN)=0:NEXT                      <138>
1440 FOR I=1 TO 999:NEXT:PRINT"{CLR}"         <000>
1450 NS=NS+1:GOTO 250                         <089>
1460 :                                        <166>
1470 REM *** INSERT ***                       <240>
1480 REM -EINFUEGEN EINER DEFINITION          <084>
1490 :                                        <196>
1500 PRINT "{CLR,DOWN,RIGHT}DEFINITIONFILE
     EINFUEGEN"                               <090>
1510 PRINT "{DOWN,RIGHT}NR. 0-5 ODER ↑"       <090>
1520 POKE 198,0:WAIT 198,1:GET A$             <174>
1530 IF A$="↑" THEN 1630                      <170>
1540 IF A$<"0" OR A$>"5" THEN 1520            <119>
1550 NI=VAL(A$):A=0                           <195>
1560 IF NI>NS THEN 1630                       <071>
1570 :P2=INT(NW(NI,A)/256):P1=NW(NI,A)-256
     *P2:P3=T(NI,A)                           <128>
1580 :POKE 49152+C,P1                         <076>
1590 :POKE 49153+C,P2                         <119>
1600 :POKE 49154+C,255-P3                     <216>
1610 A=A+1:C=C+3:IF P3<>0 THEN 1570           <233>
1620 C=C-3                                    <119>
1630 PRINT "{CLR}":GOTO 250                   <012>
1640 :                                        <092>
1650 REM *** COPY ***                         <008>
1660 REM -KOPIEREN VON NOTENREIHEN            <158>
1670 :                                        <122>
1680 PRINT "{CLR,DOWN,RIGHT}COPY"             <187>
1690 INPUT "{DOWN,RIGHT}ERSTE NOTE{2SPACE}
     :";A$                                    <039>
1700 A=VAL (A$)                               <054>
1710 INPUT "{RIGHT}LETZTE NOTE :";B           <204>
1720 A=INT(A):B=INT(B)                        <235>
1730 IF A<1 OR B<1 THEN 240                   <065>
1740 X=1:IF A>B THEN X=-1                     <043>
1750 IF A>C/3 OR B>C/3 THEN 240               <139>
1760 IF A<B THEN A=A-X:B=B-X                  <231>
1770 IF A>B THEN A=A+X:B=B+X                  <204>
1780 :                                        <232>
1790 FOR I=A TO B STEP X                      <080>
1800 :POKE 49152+C,PEEK(49152+I*3)            <026>
1810 :POKE 49153+C,PEEK(49153+I*3)            <041>
1820 :POKE 49154+C,PEEK(49154+I*3)            <056>
1830 :C=C+3                                   <012>
1840 NEXT                                     <072>
1850 GOTO 240                                 <072>
1860 :                                        <058>
1870 REM *** LOAD ***                         <240>
1880 REM -LADEN VON MUSIKSTUECKEN UND $       <252>
1890 REM SYS 53027                            <157>
1900 PRINT "{CLR,DOWN,RIGHT}LOAD SONG ODER
     DIRECTORY"                               <154>
1910 INPUT "{DOWN,RIGHT}TITEL :";A$           <227>
1920 IF A$="$" THEN 2150                      <019>
1930 IF A$="↑" THEN 240                       <252>
1940 INPUT "{DOWN,RIGHT}ANHAENGEN (J/N) :"
     ;B$                                      <044>
1950 IF B$="↑" THEN 240                       <024>
1960 OPEN 1,8,0,A$:GOSUB 2380:IF E<>0 THEN
     2120                                     <231>
1970 OPEN 1,8,0,A$                            <241>
1980 GET#1,C$,C$                              <037>
1990 IF B$<>"J" AND B$<>"JA" THEN C=0         <003>
2000 A=1                                      <139>
2010 :GET#1,C$                                <005>
2020 :A2=A1:A1=A                              <110>
2030 :IF C$<=CHR$(0) THEN A=0:GOTO 2050       <178>
2040 :A=ASC(C$)                               <028>
2050 :POKE 49152+C,A                          <015>
2060 :C=C+1                                   <243>
2070 :IF A=0 AND A1=A AND A2=A THEN 2090      <167>
2080 :GOTO 2010                               <104>
2090 C=C-3                                    <081>
2100 CLOSE 1                                  <079>
2110 GOSUB 2380                               <091>
2120 PRINT"{DOWN,RIGHT}";E;Z$:PRINT"{DOWN,
     RIGHT}TASTE DRUECKEN "                   <073>
2130 POKE 198,0:WAIT 198,1:POKE 198,0:GOTO
     240                                      <117>
2140 :                                        <084>
2150 REM --- DIRECTORY ANZEIGEN ---           <053>
2160 :                                        <104>
2170 OPEN 1,8,0,"$":KV=0                      <184>
2180 FOR I=0 TO 30:GET#1,A$:B$=B$+A$:NEXT     <139>
2190 PRINT"{2SPACE}";B$                       <064>
2200 GET#1,A$,B$,E$,F$,G$:FOR I=0 TO 26:GE
     T#1,D$:C$=C$+D$:NEXT                     <001>
2210 IF F$<=CHR$(0) THEN A=0:GOTO 2230        <166>
2220 A=ASC(F$)                                <147>
2230 IF G$<=CHR$(0) THEN B=0:GOTO 2250        <226>
2240 B=ASC(G$)                                <235>
2250 D=A+256*B                                <167>
2260 IF C$="GGGGGGGGGGGGGGGGGGGGGGGGGGG" T
     HEN C$="":GOTO 2340: REM -27 SHIFT G-    <146>
2270 PRINT D;C$:C$=""                         <063>
2280 KV=KV+1:IF KV=22 THEN 2300               <061>
2290 GOTO 2200                                <238>
2300 KV=0:PRINT "{DOWN,RIGHT}TASTE DRUECKE
     N"                                       <163>
2310 POKE 198,0:WAIT 198,1                    <252>
2320 PRINT"{UP,20SPACE}":PRINT"{3UP}"         <137>
2330 GOTO 2200                                <024>
2340 KV=0:PRINT "{DOWN,RIGHT}TASTE DRUECKE
     N"                                       <205>
2350 POKE 198,0:WAIT 198,1                    <036>
2360 CLOSE 1:POKE 198,0:GOTO 1870             <138>
2370 :                                        <060>
2380 REM +++ FEHLERKANAL ABFRAGEN +++         <071>
2390 OPEN 2,8,15                              <024>
2400 INPUT#2,E,Z$,T,S                         <026>
2410 CLOSE 1                                  <135>
2420 CLOSE 2                                  <153>
2430 RETURN                                   <202>
© 64'er
```

Musik für den C 64

Die Zeiten ändern sich. Waren vor nicht allzu langer Zeit gute Musikprogramme noch Mangelware, hat man heute die Qual der Wahl. Das Angebot ist groß.

All diese Programme haben eines gemeinsam: Sie entlocken dem Commodore 64 Töne. Der SID-Chip ist mit seinen drei Oszillatoren, den drei Hüllkurvengeneratoren, dem Filter mit den verschiedenen Filter-Modi und vor allem der Fähigkeit zur Ringmodulation und Synchronisation ein Synthesizer im Miniformat. Er steht modernen Synthesizern, die Profi-Musiker auf der Bühne einsetzen, in nichts nach. Zumindest, was die grundlegenden Fähigkeiten angeht.

Etwas schlechter ist es natürlich um den Sound bestellt. Der ist ohne Zweifel dünner als der eines professionellen Synthesizers. Verglichen mit den klanglichen Möglichkeiten anderer Home-Computer hat der Commodore 64 allerdings immer noch die Nase vorne. Nur einer übertrifft ihn, was Klänge betrifft: der Yamaha CX 5, ein MSX-Computer, der jedoch ein anderes Klangsyntheseverfahren benutzt, nämlich die sogenannte »Frequenz-Modulations-Synthese«.

Ein gutes Musikprogramm muß alle klanglichen Möglichkeiten des SID-Chips ausnutzen und trotzdem seine Programmierung zum Vergnügen werden lassen. Außerdem muß die Melodie irgendwie in den Rechner, und Musiker wollen überdies nur ungern auf alphanumerischen Tasten spielen. Nichtmusiker hingegen können meist mit Klaviertasten nicht umgehen. Sie wollen die Songs nicht live einspielen, sondern Ton für Ton eintippen. Manche lieben Noten, andere Grafik. Die Wünsche sind vielfältig. Gott sei Dank gibt es mittlerweile mindestens genauso viel Musiksoftware wie unterschiedliche Wünsche, so daß jeder das für seine speziellen Bedürfnisse Richtige finden kann.

Musicalc

Sehr viele Möglichkeiten bietet Musicalc, das in Deutschland zur Zeit nicht erhältlich ist. Das komplette System besteht aus drei Programmdisketten. Mit dem Sequenzerprogramm auf Diskette Nummer 1 gibt man eigene Melodien ein. Man kann jedoch nicht alle drei Oszillatoren gleichzeitig über die Tastatur ansprechen, sondern jeweils nur einen. Die gespielte Melodie kann man speichern. Solche Melodienspeicher nennt man Sequenzer. Der Musicalc-Sequenzer kann drei Stimmen gleichzeitig wiedergeben. Aufnehmen muß man die drei Stimmen jedoch hintereinander. Hat man eine Stimme im Kasten, nimmt man die zweite auf. Stimme Nummer 1 tönt während der Aufnahme zur Orientierung aus dem Lautsprecher. Entsprechend verfährt man bei Stimme drei. Derartige Sequenzer, in die man Melodien live einspielt, nennt man »Realtime Sequenzer«.

Wer ungern live spielt, kann die Melodien bei Musicalc auch Ton für Ton mittels spezieller Befehle eintippen. So hat man zwischen den einzelnen Eingaben Zeit zu überlegen, muß also gar nicht Klavier spielen können. In dieser Betriebsart arbeitet Musicalc als Composer. Der Musicalc-Sequenzer ist hervorragend konzipiert. Die Klangeinstellmöglichkeiten des Programmes sind sehr übersichtlich arrangiert. 32 verschiedene Songs und 32 verschiedene Klangeinstellungen kann man aus dem Arbeitsspeicher abrufen und automatisch abspielen.

Mit dem »Score Writer« auf der zweiten Diskette wandelt man die komponierten Songs in Notenschrift um. Die Notation kann man entweder auf dem Bildschirm betrachten oder auch ausdrucken. »Keyboard Maker« nennt sich das dritte Programm. Es stellt etwa 50 fertig einprogrammierte Tonleitern fremder Völker zur Verfügung. So kann jeder einen indischen Raga spielen, oder wie wär's mit chinesischen Melodien?

Für besonders Spielfaule gibt es schließlich die Musicalc-Song-Disketten. Auf jeder ist ein Set verschiedener Songs und Soundeinstellungen abgespeichert. Zum Musicalc-System gibt es auch zwei Klaviaturen. Das Colortone-Keyboard und das Colortone Pro-Keyboard. Beide konnten sich jedoch aufgrund ihres hohen Preises in Deutschland nicht durchsetzen.

Fazit: Musicalc — das Programm der beinahe unbegrenzten Möglichkeiten!

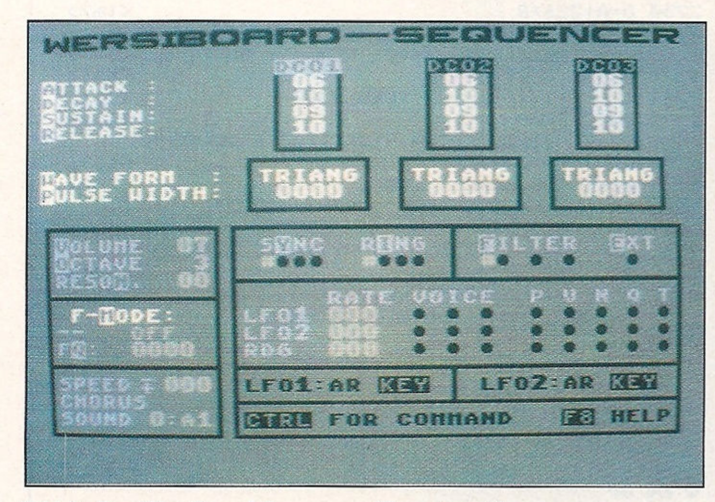


Bild 1. Der Wersiboard-Sequenzer (Hauptmenü)

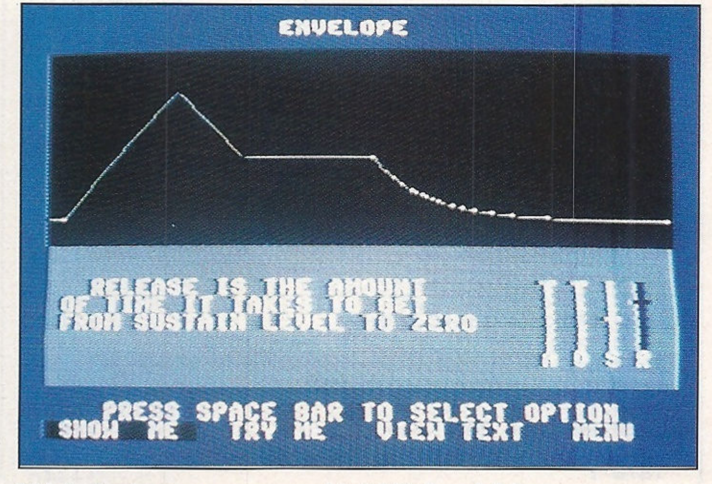


Bild 2. Kapieren, worum es geht: 3001 Sound Odyssee



Synthimat

Einstimmiges Spiel ist zwar schon etwas, aber viele wollen doch mehrstimmig auf der Tastaur spielen. Maximal können wir dem Commodore 64 natürlich nur drei verschiedene Töne gleichzeitig entlocken. So viele Oszillatoren stellt der SID-Chip zur Verfügung. Synthimat von Data Becker etwa, eines der ersten Musikprogramme, die es für den Commodore 64 gab, gestattet dreistimmiges Realtime-Spiel auf der alphanumerischen Tastatur. Die eingespielten Songs lassen sich auf die laufende Diskette aufzeichnen. Das Programm verfügt aber über keinen Composer. Auch lassen sich die eingespielten Lieder nachträglich nicht mehr ausbessern, wie zum Beispiel bei Musicalc. Hat man sich verspielt, muß man nochmal beginnen.

Synthimat bietet sehr gute Klangeinstellmöglichkeiten. Genau wie auch bei Musicalc ist für jeden Oszillator ein eigener Hüllkurvengenerator vorhanden. So lassen sich drei verschiedene Klänge gleichzeitig erzeugen. Also zum Beispiel Flötenklang für Stimme 1, Gitarre für Stimme 2 und vielleicht Rauschen als Ersatz fürs Schlagzeug auf Stimme 3. Das Mini-Orchester aus dem SID. Und jede Stimme spielt eine andere Melodie.

Fazit: Synthimat — das Low-cost-Programm für Spielernaturen und Soundfreaks.

	EXTENDED SYNTHESIZER-SYSTEM	SYNTHIMAT	ULTISYNTH	SEQUENZER 64
VCO	3	3	3	3
Kurvenform	D/S/R	D/S/R	+	—
LFO	—	8	—	2
Noise	+	+	+	+
ADSR	3	3	3	3
Pulsweite	—	+	+	+
Filtertyp	—	L/B/H + 4 Mischtypen	L/B/H	L/B/H
Filterfrequenz	—	+	—	+
Resonanz	—	+	—	—
Ringmodulator	—	+	+	+
Sync	—	+	+	+
Sychronisat extern	—	3	1	1
Anzahl Stimmen live spielbar auf Keyboard	—	3 Stimmen gleichzeitig maximal 3	Je Durchgang 1 Stimme maximal 3	Je Durchgang 1 Stimme
Composer	—	—	+	+
Songbildung (Sequenzverknüpfung)	—	—	+	+
Portamento	+	—	—	+
Notation Schirm	+	—	—	+
Hardcopy	—	—	—	+
Presents momentan verfügbar: Sounds Songs	10 1	256 1	k.A. 1	16 48 Takte
Klaviatur extern anschließbar	—	—	—	—
Preis	49,—	99.—	k.A.	39,—
Datenträger	Disk	Disk	Disk/Cass	Disk
Sonstiges	*1)	*2)	*3)	*4)

Sonstiges:
*1) Drei Stimmen werden in Realtime gleichzeitig am Bildschirm notiert.

*2) Songs werden direkt auf Diskette aufgenommen. Für jeden VCO stehen zwei separate LFOs zur Verfügung. Je ein LFO kann den VCA beziehungsweise VCF modulieren.

*3) Modulationsoszillator mit beliebigen Kurvenformen belegbar. Eine Stimme Realtime spielbar, zwei Stimmen als Playback zu programmieren.

*4) Triolen, Quintolen, Septolen spielbar, Notendruck mit Zusatzprogramm.

*5) Realtime-Sequenzer ist in Entwicklung.

*6)7 Rhythmuspresets zur Begleitung. Keine Notenschlüssel. Keine erniedrigten Töne möglich.

*7) Randomgenerator, Wah-Wah, 9 Sequenzerlinien können im Arbeitsspeicher zu Songs gekoppelt werden. Songteile können beim Abspielen von Diskette nachgeladen werden. Begleitautomatik Grafikeffekte



Bild 3. Das Wersiboard zum C 64

Wersiboard 64

Wem die alphanumerischen Tasten zu wenig musikalischen Komfort bieten, der greift am besten zu einem der Klaviatur-Systeme. Sie verwandeln den Commodore 64 in ein Musikinstrument mit schwarzen und weißen Tasten. Die Firma Wersi, Hersteller weltbekannter und beliebter Orgelsysteme, bietet für 495 Mark eine Grundsoftware plus externe Klaviatur aus Plastik. Die Vier-Oktaven-Klaviatur wird am User-Port des Commodore 64 angeschlossen. Besagte Grundsoftware besteht aus zwei Programmen, Mono 64 und Poly 64. Sie erlauben entweder monophones Spiel mit komplexen Klängen oder dreistimmiges Spiel mit weniger komfortablen Klangeinstellmöglichkeiten. Die Sounds lassen sich in der Grundversion nicht auf Diskette abspeichern.

Verschiedene Softwarezusätze erweitern die Möglichkeiten ziemlich. Sound-Pack 1 gestattet das Abspeichern von bis zu 32 Sounds im Arbeitsspeicher und Hardcopy sämtlicher Parameter. Sound-Pac 2 bietet einen komfortablen Sequenzer mit ausgefeilten Klangeinstellmöglichkeiten. Die Songs lassen sich jedoch nicht Realtime einspielen. Ein absolutes Novum bietet Sound-Pack 3. Hiermit kann man alle Tasten der Klaviatur unabhängig voneinander völlig frei stimmen. Vor allem für experimentierende Musiker ist dies interessant. So kann man etwa originalgetreu Musik fremder Völker spielen. In Asien wie auch in anderen Kulturkreisen haben die Tonleitern ganz andere Intervalle als bei uns.

Fazit: Das Wersiboard — komfortable Hard- und Software für alle, die schwarze und weiße Tasten lieben.

MUSIC MASCHINE	WERSIBOARD	KAWASAKI SYNTHESIZER	KAWASAKI RHYTHM ROCKER	3001 SOUND ODYSSEE
3	3	3	3	3
D/S/R	D/S/R	D/S/R	D/S/R	D/S/R
—	2	+	—	—
+	+—	+	+	+
3	3	3	1 (nur Decay regelbar)	3
—	+	+	—	+
—	L/B/H	L/B/H	—	L/B/H
k.A.	+	+	—	+
k.A.	—	—	—	+
—	+	+	—	+
—	+	+	—	+
k.A.	—	—	—	—
Je Durchgang 1 Stimme, maximal 2	1	3	1	1
+	+	(+)	—	—
—	+	+	—	—
—	+	—	+(Pitch bend)	—
+	+	Buchstaben-notation	—	—
—	—	—	—	—
k.A.	320	4	10 feste	100
1	1	siehe 11	1	1
—	+	incredible Keyboard	incredible Keyboard	incredible Keyboard
59,—	495,—	99,—	99,—	99,—
Cart	Disk	Cass/Disk	Cass/Disk	Cass/Disk
*5)	*6)	*7)	*8)	*9)

*8) hervorragende Grafikeffekte zum Spiel programmierbar. Automatische Melodie und Percussionbegleitung. Drei Synthesizersounds, 6 Spacesounds und Baß.
Diverse Effektmöglichkeiten. Die Begleitmelodien sind kompatibel zum Kawasaki Synthesizer.
*9) Bedienung per Joystick. Softwarelehrbuch, Spielautomatik.
Info: Kawasaki Synthesizer, Kawasaki Rhythm Performer, 3001 Sound Odyssee, incredible Keyboard: Ariolasoft, Postfach 1350, 4830 Gütersloh
Wersiboard: Wersi Orgel- und Piano-Bausätze, Industriestr., 5401 Helsenbach
Synthimat: Data Becker, Merowingerstr. 31, 4000 Düsseldorf 1
Extended Synthesizer System: Interface Age, Josefsburgstr. 6, 8000 München 80
Ulti Synth: Micro-Händler, Robert-Koch-Str. 1, 4050 Mönchengladbach 1
Sequenzer 64, Music Machine: Commodore, 6000 Frankfurt/Main 71. Ebenfalls über Commodore-Händler zu beziehen: Gitarre 64 (39 Mark), Musik Maker (89 Mark) und Musik Composer (59 Mark)

Wer seinem Commodore 64 mehr als drei Töne gleichzeitig entlocken will, für den kommt nur das Musipack 64-System der italienischen Firma Jen in Frage. Auch hier gehört eine externe Klaviatur mit einem Bereich von vier Oktaven zur Grundausstattung. Darüber hinaus ist im Expansion-Portstecker ein zusätzlicher Sound-Chip untergebracht.

Jen Musipack C 64

Dieser generiert weitere Stimmen. Insgesamt stehen acht (!) voneinander unabhängige Stimmen zur Verfügung. Dafür sind die klanglichen Möglichkeiten etwas eingeschränkt. Nicht alle Parameter des SID-Chips lassen sich nutzen. Der Realtime-Sequenzer gestattet die Abspeicherung bis zu sechsstimmiger Musikstücke. Die verbleibenden Stimmen kann man live auf der Klaviatur zuspielen. Die Software ist fest auf einem PROM gespeichert. Langwieriges Laden von der Diskette entfällt deshalb. Leider wird das Jen Musipack in Deutschland zur Zeit nicht mehr vertrieben.

Fazit: Musipack C 64 — Mit acht Stimmen in den Himmel der Polyphonie.

Und wie steht's mit den Noten? Das Extended Synthesizer-System bringt sie auf den Bildschirm! Und zwar ohne langwierige Umwandlungsprozedur, wie sie zum Beispiel bei Musicalc nötig ist. Hier sausen alle drei Stimmen in Notenform Realtime über den Bildschirm. Die zugehörige Musik klingt aus dem Lautsprecher. Der Preis für die Noten am Bildschirm: weniger klangliche

Extended Synthesizer-System

Möglichkeiten. Auch hier lassen sich nicht alle Parameter des SID-Chips nutzen. Dafür kann jeder einzelne Ton einer Melodie mit eigenem Sound versehen werden. Soundwechsel sind also blitzschnell möglich. Das schafft nur der Extended Synthesizer! Extended Synthesizer ist ein reines Composer-Programm. Das heißt, die Melodien gibt man Ton für Ton mittels spezieller Befehle ein. Realtime lassen sich hier überhaupt keine Melodien spielen.

Fazit: Extended Synthesizer, das Programm für Komponisten und Freunde guter Noten.

Kawasaki Synthesizer

Was nutzen die schönsten Töne, wenn sich auf dem Bildschirm nichts ansprechendes tut. Man denke an die vielen Video-Clips. Was wären Sie ohne Bild? Konzertierende SID-Musiker benötigen Software, die passende Bilder zum Ton liefern. Die Firma Sight & Sound bietet für diesen Zweck das Passende. Live Konzert plus Realtime-Grafik: Der Videoclip im Wohnzimmer. Klar, daß sich dies nur ein Profimusiker ausdenken konnte. Ryo Kawasaki, der Softwareautor, ist gebürtiger Japaner, lebt in USA. Von Beruf ist er Popstar und Software-Künstler. Mit dem nach ihm benannten Kawasaki Synthesizer kann man die ausgefallensten Klänge einfach einstellen und speichern. Über die alphanumerische Tastatur kann man live spielen. Auch ein dreistimmiger Sequenzer ist vorhanden.

In Teil 2 des Programmes geht die Show dann richtig los. Die Space Travel-Show beginnt. Eine Klaviatur schwebt durchs All, Roboter hüpfen im Takt auf und ab — der Robot Dance auf dem Computer-Monitor. Dazu die Musik — entweder gespielt auf der Computer-Tastatur: drei Stim-

men live — oder: Wir überlassen Baß und Schlagzeug dem Computer und spielen nur noch eine Stimme live hinzu. Wem das auch noch zu viel ist, der verdammt den Commodore 64 total zur Musikbox und legt einen der mitgelieferten 30 Software-Demosongs auf. Das Spektrum reicht vom Calypso zum Raga, vom Honk über Fuge und Präludium bis hin zum 13/8 Baß.

Fazit: Der Kawasaki Synthesizer, das Programm für SID-Künstler mit Konzert-Ambitionen.

Kawasaki Rhythm Rocker

Mehr Grafikmöglichkeiten, aber weniger eigene Kreativität im musikalischen Sinne läßt der Kawasaki Rhythm Rocker zu. Dafür macht er auch den blutigsten Musik-Laien zum Popstar. Man muß nicht viel Ahnung von Musik haben. Sofern man die Tasten des Commodore 64 trifft hat man schon gewonnen. Der Kawasaki Rhythm Rocker greift kräftig unter die Arme. Er unterstützt das »Spiel« mit automatischen Playbacks, diversen, festeinprogrammierten Schlagzeugrhythmen und Sounds vom Jet bis hin zu diversen Space guns und sonstigen Effekten der Video-game-scene. Dazu entsteht Computer-Grafik auf dem Bildschirm. Farbe, Form und Bewegung. Als Hintergrund: wählbar entweder Koordinatennetz und Sternhimmel oder Comic-Klaviatur mit Lauflicht.

Fazit: der Kawasaki Rhythm Rocker macht selbst Nichtmusiker zu Stars mit eigenem Videoclip.

3001 Sound Odyssee

Doch auf solchen, eigentlich unverdienten Lorbeeren sollte man sich nicht ausruhen. Deshalb bietet Sight and Sound für all die grünen Jungs ein tönendes Software Synthesizer-»Lehrbuch«: 3001 Sound Odyssee. Es klärt alle Geheimnisse um Hüllkurven, Filter, Ringmodulation, Synchronisation und all die anderen Fachausdrücke der Synthesizertechnik leichtverständlich auf. Per Joystick kommandiert man die einzelnen Lektionen. Auf dieser Sound-Odyssee kapiert man schneller als man denkt, worum es eigentlich geht. Neben dem Lehrbuch ist noch ein Synthesizerprogramm auf der Diskette, der Microsynth. Er bietet alle wichtigen Klangeinstellmöglichkeiten und eine recht ordentliche Akkordautomatik.

Fazit: 3001 Sound Odyssee — Die Geheimnisse der Klangsynthese und Synthesizertechnik Schritt für Schritt kapiert.

The incredible Keyboard

Will man nicht viel Geld investieren, aber trotzdem auf schwarzen und weißen Tasten spielen, sollte man sich das incredible Keyboard der Firma Sight and Sound zulegen. Eine Mini-Plastik-Klaviatur zum Aufstecken auf den Commodore 64. Man kann sie jedoch nur in Verbindung mit einem Grundprogramm und zwei Notenbüchlein erwerben. Die zugehörige Software bietet keine großartigen Soundeinstellmöglichkeiten. Sie wandelt den Commodore 64 lediglich in eine dreistimmig spielbare Orgel. Die »Clip on«-Tastatur ist kompatibel zu allen anderen Sight and Sound-Programmen.

Fazit: Das incredible Keyboard — Miniklaviatur und Orgelprogramm für genügsame Tastenliebhaber.

Sound Sampler für den Commodore 64

Ein besonderes High-Light: der Low-Cost-Sound Sampler der Firma Musicsales Limited. Wer wollte nicht immer schon mal gerne klein Bello in den Computer sperren, nachdem er eine Stunde im Hof kläfft? Der Sound Sampler macht's möglich. Zumindest was den Kläffer betrifft. Natürliche Klänge im Computer. Spielbar in sämtlichen Tonlagen per alphanumerischer Tastatur.

Aus Kirchenglocken werden so nepalesische Tempelglöckchen, aus Bello's Kläffer ein brüllendes Ungetüm. Klänge müssen somit nicht mehr in langwieriger Prozedur synthetisiert werden. Man fängt sie einfach per Mikrofon ein und speichert sie im Computer ab. Auf Klangpirsch mit Mikrofon und Computer! Bis zu vier Klänge finden gleichzeitig im Arbeitsspeicher Platz. Sie lassen sich vor- und rückwärts abspielen sowie auf verschiedene Art grafisch am Bildschirm darstellen.

Der Sound Sampler kann Klänge auch Realtime, also direkt bei der Eingabe, verändern. So erzeugt man sehr interessante Klangverfremdungen (Echo, Harmonizer). Wer nun glaubt, das müsse unheimlich teuer sein, der irrt. Der Sound Sampler kostet zirka 199 Mark. Einziges Manko, bis jetzt gibt es ihn noch nicht zu kaufen.

Fazit: Der Sound Sampler — das Programm für alle, die das Besondere lieben.

Music Maker, Playalong Album und Prosynth

Dieselbe Firma bietet drei weitere Musikprogramme zum Commodore 64 an: Music Maker, Playalong Album und Prosynth (Preis voraussichtlich je um zirka 50 Mark). Genau wie auch der Sound Sampler verfügen sie über ein besonderes Feature: Sie sind midi-kompatibel. Musiker benutzen das Midi-System, um mehrere Synthesizer und ein Elektronik-Schlagzeug vom Computer aus anzusteuern. Auf diese Weise spielen alle Geräte im Gleichtakt. Der SID-Chip unseres Commodore 64 kann in diesen Kanon taktvoll einstimmen, sofern man eines der besagten Programme nutzt.

Fazit: Music Maker, Playalong Album und Prosynth — die Programme für Musikfreaks mit Midi-Ambitionen.

Gitarre 64

Warum sollen eigentlich immer nur Keyboarder dem Commodore 64 Töne entlocken, dachte sich Werner Kracht, Musiksoftware-Spezialist und Gitarrist aus Hamburg. Gesagt — getan. Gitarre 64 war geboren. Dieses Programm wird von der Firma Commodore vertrieben. Es bietet enorme Hilfen für Gitarristen. Umständliches Nachschlagen von Gitarrengriffen in Grifftabellen kann man sich damit ab sofort sparen. Der Computer berechnet die wichtigsten Griffe und stellt sie am Bildschirm samt Gitarrengriffbrett dar.

Weiterhin lassen sich unbekannte Griffe analysieren und passende Tonleitern hierzu finden. Umgekehrt findet man schnell zu einer bestimmten eingegebenen Melodie die passende Gitarrenbegleitung.

Fazit: Gitarre 64 — ein Muß für computerorientierte Gitarristen!

Damit sind wir am Ende unserer kleinen Exkursion durch die Welt der Commodore 64-Musikprogramme angelangt. Die besprochenen Beispiele sollen exemplarisch für viele weitere stehen und unterschiedliche Programmstrukturen und Konzepte etwas verdeutlichen. Vielleicht fällt die Auswahl eines passenden Musikprogramms nun etwas leichter. (Richard Aicher/ev)

Sound Master

Musik und Soundeffekte auf dem C 64 zu programmieren war bisher sehr kompliziert, kam man doch nicht um den POKE-Befehl herum. Unsere Basic-Erweiterung Sound Master behebt dieses Übel.

Das Programm Sound Master (Listing 1) muß mit dem MSE eingegeben werden. Später kann es dann mit »LOAD "SOUND MASTER",8« und »RUN« gestartet werden. Danach stehen Ihnen acht neue Basic-Befehle zur Verfügung.

Zum Ausprobieren (nach dem Starten von Listing 1) sind noch drei Demoprogramme, Listings 2 bis 4, abgedruckt. Diese können aufgrund der neu definierten Basic-Befehle nicht mit dem Checksummer eingegeben werden! Nun zur Beschreibung der acht neuen Befehle:

SOUNDCLEAR

Initialisiert alle Tonregister neu (Ton aus; Frequenz 0 etc.).

VOLUME vo

Bestimmt die Gesamtlautstärke (vo). vo kann Werte von 0 bis 15 annehmen. Zum Beispiel Volume 15 = volle Lautstärke.

Im folgenden gibt der Parameter tg immer den gewünschten Tongenerator an. Es stehen drei Tongeneratoren zur Verfügung (1 bis 3), die gleichzeitig gespielt werden können. Um einen Ton zu erzeugen, müssen zuerst mit den folgenden Befehlen die Tonparameter bestimmt werden. Mit PLAY wird der Ton dann eingeschaltet.

OSCILLATE tg,fq

Bestimmt die Frequenz (fg) von Tongenerator tg. Die Frequenz kann direkt in Hertz angegeben werden! Die maximale Frequenz liegt bei 3848 Hz. Die Genauigkeit beträgt mehrere Stellen nach dem Komma. Zum Beispiel Oscillate 1,1000 = Tongenerator 1 schwingt mit 1000 Hz.

TUNE tg,no$

Eine andere Möglichkeit, die Tonfrequenz zu bestimmen ist die Angabe einer Note (no$). Diese muß in einem String stehen. Die Syntax ist die gleiche wie in Anhang P des 64-Handbuchs beschrieben, das heißt Buchstabe der Note — Oktave. Der Bindestrich kann weggelassen werden. Die Programmierung eines Musikstücks mit dem TUNE-Befehl ist sehr gut in Listing 4 demonstriert. Zum Beispiel »TUNE 2,"D4" « spielt in TG 2 die Note D, Oktave 4; »TUNE 3,A$« spielt die Note aus A$ in TG 3

Anmerkung zu den OSCILLATE- und TUNE-Befehlen:

Die Umrechnung der Frequenz beziehungsweise der Note erfolgt auf Basis der deutschen PAL-Version des Commodore 64 (Prozessortakt 985 kHz). Bei der amerikanischen NTSC-Version ist eine andere Umrechnung erforderlich (Prozessortakt 1022 kHz). (In der Tabelle im Handbuch sind übrigens die amerikanischen Werte für die Noten vermerkt, auf einem deutschen 64 wird die Note also anders als beim Sound Master nicht genau getroffen!)

ENVELOPE tg,a,d,s,r

Dieser Befehl bestimmt die Hüllkurve des Tons.

a = Anstiegszeit
d = Abfallzeit
s = Haltepegel
r = Ausklingzeit

Es können Werte von 0 bis 15 eingegeben werden.

WAVE tg,wf,(tv),ef ...

Bestimmt die Wellenform (wf), mit der der Ton gespielt wird.

1 = Dreieck
2 = Sägezahn
3 = Rechteck
4 = Rauschen

Wird Rechteck ausgewählt, muß zusätzlich eine Angabe des Tastverhältnisses (tv) folgen (0 bis 100 Prozent). Die Genauigkeit betätigt hier mehrere Stellen nach dem Komma.
Effekte: Nach diesem Parameter können noch einzelne Buchstaben folgen, um bestimmte Toneffekte zu programmieren.
OFF: Waren die Angaben zur Wellenform für Stimme 3 bestimmt (tg = 3), so kann mit OFF diese (aus technischen Gründen leider nur diese) Stimme stumm geschaltet werden.

S: Der ausgewählte Tongenerator (tg) wird mit einem anderen Tongenerator synchronisiert, und zwar 1 mit 3, 2 mit 1 und 3 mit 2 (siehe Listing 3).

R: Die Dreieckspannung des ausgewählten Tongenerators (tg) wird durch eine Ringmodulation zweier Tongeneratoren ersetzt (1 mit 3, 2 mit 1 und 3 mit 2) (siehe Listing 2).

Zum Beispiel »WAVE 2,2« : Stimme 2 spielt mit Sägezahn.
»WAVE 3,1OFF« : Stimme 3 spielt mit Dreieck, wird jedoch stumm geschaltet. Man hört also nichts. Dies braucht man zum Beispiel, wenn der Ton von Stimme 3 bei Synchronisation oder Ringmodulation nur stören würde.

PLAY, tg,td

Mit diesem Befehl wird der Ton eingeschaltet (td ist die Tondauer; maximal 65535, das sind zirka 46 Sekunden). Anstatt eine Tondauer anzugeben, kann mit »PLAY tg,ON« der Ton nur eingeschaltet werden. Das Basic-Programm wird dann sofort fortgeführt. Mit »PLAY tg,OFF« wird der Ton wieder ausgeschaltet.
»PLAY tg,ON«: Schaltet tg an

FILTER fh,fq,re,tg....

Der SID bietet die Möglichkeit, eine oder mehrere Stimmen über einen Filter zu leiten. fh steht für einen einzelnen Buchstaben und dient zur Auswahl der gewünschten Filterart.

H = Hochpaßfilter
T = Tiefpaßfilter
B = Bandpaßfilter
N = Notchfilter

Dann folgt die Filterfrequenz (fq) in Hertz (30 bis 11800 Hz).

Die Resonanz (re) hebt die Frequenzen in der Nähe der Filterfrequenz hervor (Werte linear 0 bis 15). Nun folgen die Nummern der Tongeneratoren, die über den Filter geleitet werden sollen (4 = Audioeingang).

Zum Beispiel »FILTER H,2000,10,1, 2« : Stimme 1 und 2 werden über den Filter geleitet. Filterfrequenz = 2000 Hz, Resonanz = 10, Hochpaß eingeschaltet.

Nun noch ein kurzes Beispiel, wie man einen einfachen Ton programmiert.

```
VOLUME 15:REM Volle Lautstärke
ENVELOPE 1,0,0,15,0:REM nur »Halten« auf voller Lautstärke
WAVE 1,2:REM Sägezahn
OSCILLATE 1,500:REM 500 Hertz
PLAY 1,1000:REM Spielt Ton
oder PLAY 1,ON:REM TG 1 an
FOR I=0TO1000:OSCILLATE 1,I:
NEXT:REM Als Frequenz werden Werte von 0 bis 1000 eingesetzt. Es entsteht ein ansteigender Ton.
PLAY 1,OFF:REM Ton aus
```

Alle Befehle sind nach dem Programmstart fest ins Basic eingebunden und bleiben bis zum Ausschalten erhalten.
Alle Befehle können normal abgekürzt werden, zum Beispiel »S + Shift—O« = SOUNDCLEAR.

(P. Menke/bs)

```
programm : sound master  0801 0e64
-----------------------------------
0801 : 28 08 c0 07 9e 32 31 31  e1
0809 : 35 3a 20 4d 42 53 32 20  d5
0811 : 20 3c 2a 53 4f 55 4e 44  a6
0819 : 20 4d 41 53 54 45 52 2a  a8
0821 : 3e 20 20 20 20 20 00 00  7f
0829 : 00 00 00 00 20 2a 20 20  3e
0831 : 4d 50 52 47 2e 42 41 53  c4
0839 : 2e 53 54 41 52 54 2e 47  5d
0841 : 8a 08 ae 4a 08 ac 4b 08  e7
0849 : ad 63 0e 8d d9 c5 ec 41  df
0851 : 08 d0 08 cc 42 08 d0 03  0b
0859 : 4c 77 08 ce 4a 08 ce 4d  f7
0861 : 08 a9 ff cd 4a 08 d0 03  26
0869 : ce 4b 08 cd 4d 08 d0 d2  97
0871 : ce 4e 08 4c 43 08 a9 00  0d
0879 : 8d 01 08 8d 02 08 a9 01  44
0881 : 85 2d a9 08 85 2e 4c 00  03
0889 : c0 4c 24 c0 00 00 00 00  91
0891 : 00 00 00 00 00 00 00 00  92
0899 : 00 00 00 00 00 00 00 00  9a
08a1 : 00 00 00 00 00 00 00 00  a2
08a9 : 00 00 00 00 00 a9 53 8d  5f
08b1 : 04 03 a9 c0 8d 05 03 a9  1a
08b9 : 51 8d 08 03 a9 c1 8d 09  24
08c1 : 03 a9 1c 8d 06 03 a9 c1  f4
08c9 : 8d 07 03 20 48 c0 4c 0e  77
08d1 : c5 a0 1c a9 00 99 04 c0  81
08d9 : 88 10 fa 60 a6 7a a0 04  fd
08e1 : 84 0f bd 00 02 10 07 c9  ad
08e9 : ff f0 44 e8 d0 f4 c9 20  ab
08f1 : f0 3d 85 08 c9 22 f0 5b  0a
08f9 : 24 0f 70 33 c9 3f d0 04  09
0901 : a9 99 d0 2b c9 30 90 04  79
0909 : c9 3c 90 23 84 71 a0 00  cf
0911 : 84 0b 88 86 7a ca c8 e8  01
0919 : bd 00 02 38 f9 9e a0 f0  57
0921 : f5 c9 80 d0 35 05 0b c9  70
0929 : 8b d0 02 a9 cd a4 71 e8  6c
0931 : c8 99 fb 01 c9 00 f0 38  b6
0939 : 38 e9 3a f0 04 c9 49 d0  68
0941 : 02 85 0f 38 e9 55 d0 9a  92
0949 : 85 08 bd 00 02 f0 e0 c5  f8
0951 : 08 f0 dc c8 99 fb 01 e8  71
0959 : d0 f0 a6 7a e6 0b c8 b9  f8
0961 : 9d a0 10 fa b9 9e a0 d0  66
0969 : af f0 0f bd 00 02 10 bd  d8
0971 : 99 fd 01 c6 7b a9 ff 85  32
0979 : 7a 60 a0 00 b9 78 c1 d0  53
0981 : 02 c8 e8 bd 00 02 38 f9  be
0989 : 78 c1 f0 f5 c9 80 d0 04  c8
0991 : 05 0b d0 99 a6 7a e6 0b  73
0999 : c8 b9 77 c1 10 fa b9 78  05
09a1 : c1 d0 e0 f0 c6 10 0f 24  92
09a9 : 0f 30 0b c9 ff f0 07 c9  03
09b1 : cc b0 06 4c 24 a7 4c f3  79
09b9 : a6 38 e9 cb aa 84 49 a0  a4
09c1 : ff ca f0 08 c8 b9 78 c1  22
09c9 : 10 fa 30 f5 c8 b9 78 c1  e1
09d1 : 30 05 20 47 ab d0 f5 4c  26
09d9 : ef a6 20 73 00 20 5a c1  80
09e1 : 4c ae a7 c9 cd 90 04 c9  ac
09e9 : d7 90 06 20 79 00 4c ed  33
09f1 : a7 38 e9 cd 0a aa bd b6  43
09f9 : c1 48 bd b5 c1 48 4c 73  7b
0a01 : 00 4f 46 c6 49 c6 56 4f  d6
0a09 : 4c 55 4d c5 57 41 56 c5  70
0a11 : 45 4e 56 45 4c 4f 50 c5  c8
0a19 : 4f 53 43 49 4c 4c 41 54  e1
0a21 : c5 54 55 4e c5 50 4c 41  c2
0a29 : d9 46 49 4c 54 45 d2 53  63
0a31 : 4f 55 4e 44 43 4c 45 41  75
0a39 : d2 48 45 4c d0 00 92 c5  ed
0a41 : c8 c1 1d c2 bc c2 f5 c2  c9
0a49 : 7b c3 18 c3 40 c4 07 c5  f6
0a51 : b6 c5 20 9e b7 e0 10 b0  ea
0a59 : 32 8e 03 c0 ad 1c c0 29  bc
0a61 : f0 0d 03 c0 8d 1c c0 4c  06
0a69 : e6 c2 98 48 20 fd ae 4c  25
0a71 : eb c1 98 48 20 9e b7 e0  04
0a79 : 10 b0 0f 68 a8 8a 60 01  15
0a81 : 02 04 08 10 20 40 80 04  98
0a89 : 0b 12 68 a2 0e 4c 37 a4  75
0a91 : 20 9e b7 ca e0 03 b0 f3  18
0a99 : 8e 23 c0 bd fe c1 85 fe  b3
0aa1 : a9 c0 85 ff 4c fd ae 20  bb
0aa9 : 07 c2 20 9e b7 ca e0 04  4b
0ab1 : b0 d9 8a 09 04 aa a0 04  32
0ab9 : b1 fe 29 09 1d f6 c1 91  09
0ac1 : fe e0 06 d0 22 20 fd ae  43
0ac9 : 20 8a ad a9 84 a0 c2 20  67
0ad1 : 28 ba 20 f7 b7 a5 15 c9  ee
0ad9 : 10 90 03 4c 02 c2 a0 03  3a
0ae1 : 91 fe 88 a5 14 91 fe ae  ef
0ae9 : 23 c0 e0 02 d0 08 ad 1c  21
0af1 : c0 29 7f 8d 1c c0 20 79  13
0af9 : 00 f0 74 20 fd ae c9 52  b4
0b01 : f0 10 c9 53 f0 1a c9 cc  77
0b09 : f0 24 4c 77 c3 86 23 cc  a4
0b11 : cc cd a0 04 b1 fe 09 04  ac
0b19 : 91 fe 20 73 00 4c 70 c2  4a
0b21 : a0 04 b1 fe 09 02 91 fe  f4
0b29 : 20 73 00 4c 70 c2 ae 23  ab
0b31 : c0 e0 02 f0 03 4c 02 c2  20
0b39 : ad 1c c0 09 80 8d 1c c0  ac
0b41 : 20 73 00 4c 70 c2 20 07  50
0b49 : c2 20 e9 c1 0a 0a 0a 0a  fb
0b51 : a0 05 91 fe 20 e1 c1 11  f2
0b59 : fe 91 fe 20 e1 c1 0a 0a  4c
0b61 : 0a 0a c8 91 fe 20 e1 c1  d1
0b69 : 11 fe 91 fe 4c e6 c2 20  85
0b71 : ea c2 60 a0 1d b9 03 c0  16
0b79 : 99 ff d3 88 d0 f7 60 20  a6
0b81 : 07 c2 20 8a ad a9 14 a0  fc
0b89 : c3 20 28 ba 20 f7 b7 a5  aa
0b91 : 14 a0 00 91 fe c8 a5 15  1e
0b99 : 91 fe 4c e6 c2 85 08 3a  86
0ba1 : 33 be 20 07 c2 20 79 00  2f
0ba9 : 90 22 c9 91 f0 0a c9 cc  0f
0bb1 : d0 4e 20 73 00 4c 69 c3  ae
0bb9 : 20 73 00 20 38 c3 4c e6  38
0bc1 : c2 a0 04 b1 fe 09 01 91  6a
0bc9 : fe 4c ea c2 20 8a ad 20  4e
0bd1 : f7 b7 20 38 c3 a2 00 a5  50
0bd9 : 14 d0 04 a5 15 f0 13 ca  c6
0be1 : d0 fd 38 a5 14 e9 01 85  12
0be9 : 14 a5 15 e9 00 85 15 4c  6b
0bf1 : 4e c3 a0 04 b1 fe 29 fe  7f
0bf9 : 91 fe 20 ea c2 4c e6 c2  1e
0c01 : a2 0b 4c 37 a4 20 07 c2  10
0c09 : 20 9e ad 20 a3 b6 a0 01  5c
0c11 : b1 22 8d 21 c0 88 b1 22  b6
0c19 : 8d 22 c0 c8 a2 ff e8 e0  90
0c21 : 0c b0 dd dd 17 c4 d0 f6  81
0c29 : ad 21 c0 c9 23 d0 0f e8  97
0c31 : ad 22 c0 dd 17 c4 d0 c8  48
0c39 : c8 b1 22 8d 21 c0 ad 21  25
0c41 : c0 c9 2d d0 06 c8 b1 22  fd
0c49 : 8d 21 c0 8a 0a aa bd 23  1b
0c51 : c4 8d 03 c0 e8 bd 23 c4  47
0c59 : 8d 22 c0 ad 21 c0 38 e9  aa
0c61 : 30 c9 08 90 03 4c 02 c2  aa
0c69 : 49 07 d0 0b a0 00 b1 22  e0
0c71 : c9 48 d0 1b 4c 02 c2 aa  2b
0c79 : a0 00 b1 22 c9 48 d0 06  f8
0c81 : 38 6e 22 c0 30 03 4e 22  2a
0c89 : c0 6e 03 c0 ca d0 f7 ad  c8
0c91 : 03 c0 a0 00 91 fe ad 22  28
0c99 : c0 a0 01 91 fe 4c e6 c2  8f
0ca1 : 43 43 44 44 45 46 46 47  4e
0ca9 : 47 41 41 48 38 8b 81 93  f7
0cb1 : 45 9c 90 a5 68 af d6 b9  f0
0cb9 : e4 c4 99 d0 ff dc 24 ea  cd
0cc1 : 10 f8 cf 06 20 73 00 4c  38
0cc9 : e6 c2 20 f4 c4 20 79 00  ea
0cd1 : c9 48 f0 19 c9 54 f0 0f  3f
0cd9 : c9 42 f0 0e c9 4e f0 10  b4
0ce1 : c9 cc f0 e0 4c 77 c3 a9  4b
0ce9 : 10 2c a9 20 2c a9 40 2c  e7
0cf1 : a9 50 0d 1c c0 8d 1c c0  f4
0cf9 : 20 73 00 20 fd ae 20 8a  c2
0d01 : ad a9 e5 a0 c4 20 5b bc  44
0d09 : 10 05 f0 03 4c 02 c2 a9  6c
0d11 : ea a0 c4 20 67 b8 a9 ef  43
0d19 : a0 c4 20 28 ba 20 f7 b7  24
0d21 : a5 15 c9 08 90 03 4c 02  1b
0d29 : c2 a5 14 29 07 8d 19 c0  ab
0d31 : a5 14 4a 4a 4a 8d 1a c0  b7
0d39 : a5 15 0a 0a 0a 0a 0a 0d  60
0d41 : 1a c0 8d 1a c0 20 e1 c1  7a
0d49 : 0a 0a 0a 0a 8d 1b c0 20  11
0d51 : fd ae 20 9e b7 ca e0 04  df
0d59 : 90 03 4c 02 c2 bd f6 c1  38
0d61 : 0d 1b c0 8d 1b c0 20 79  09
0d69 : 00 d0 e4 4c e6 c2 85 70  0f
0d71 : 00 00 00 85 f0 00 00 00  31
0d79 : 7e 30 8d 3d 96 a9 00 8d  ec
0d81 : 19 c0 8d 1a c0 8d 1b c0  07
0d89 : ad 1c c0 29 0f 8d 1c c0  e9
0d91 : 60 20 48 c0 4c e6 c2 a9  86
0d99 : 1f 85 fe a9 c5 85 ff 20  38
0da1 : 78 c5 20 44 a6 4c 74 a4  74
0da9 : 93 11 11 1d 1d 1d 2a 2a  65
0db1 : 2a 2a 2a 20 53 4f 55 4e  21
0db9 : 44 20 4d 41 53 54 45 52  1a
0dc1 : 20 2a 2a 2a 2a 2a be be  33
0dc9 : 1d 1d 1d 20 57 52 49 54  96
0dd1 : 54 45 4e 20 42 59 20 50  70
0dd9 : 45 54 45 52 20 4d 45 4e  02
0de1 : 4b 45 20 be be 1d 1d 1d  32
0de9 : 33 38 39 31 31 20 42 41  4c
0df1 : 53 49 43 20 42 59 54 45  89
0df9 : 53 20 46 52 45 45 be be  2f
0e01 : bf a0 00 b1 fe c9 bf f0  66
0e09 : 12 c9 be d0 02 a9 0d 20  ab
0e11 : d2 ff e6 fe d0 02 e6 ff  35
0e19 : 4c 78 c5 60 20 9e ad 20  0d
0e21 : 79 00 c9 89 f0 05 a9 a7  6b
0e29 : 20 ff ae a5 61 d0 06 20  9e
0e31 : 09 a9 4c fb a8 20 79 00  13
0e39 : b0 03 4c a0 a8 4c 5a c1  6c
0e41 : d0 1d a9 0d 20 d2 ff a0  86
0e49 : 00 b9 7d c1 f0 10 10 07  9b
0e51 : 29 7f 20 d2 ff a9 0d 20  5e
0e59 : d2 ff c8 4c c0 c5 60 4c  3b
0e61 : 77 c3 a7 96 ff 20 a5 ff  0e
```

Listing 1. Die Basic-Erweiterung Sound Master muß mit dem MSE abgetippt werden (siehe Seite 54)

```
100 REM DEMONSTRATION
110 REM FUER RINGMODULATION
120 :
130 SOUNDCLEAR:VOLUME 15
140 ENVELOPE 1,0,0,15,0
150 ENVELOPE 3,0,0,15,0
160 :
170 WAVE 1,1,R  :REM RINGMODULATION
180 WAVE 3,3,30
190 :
200 PLAY 3,ON
210 PLAY 1,ON
220 :
230 FOR I=1 TO 3000 STEP 100
240 FOR X=1 TO 3000 STEP 50
250 OSCILLATE 3,X
260 NEXT
270 OSCILLATE 1,I
280 NEXT
290 :
300 SOUNDCLEAR
READY.
```

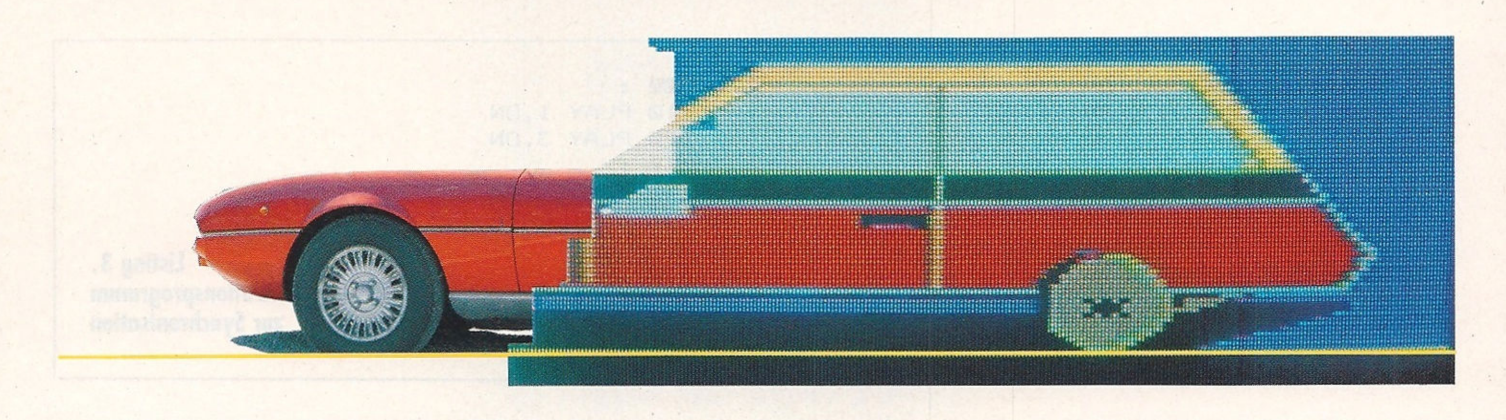
Listing 2. Ein Demonstrationsprogramm zur Ringmodulation

```
100 REM DEMONSTRATION
110 REM FUER SYNCHRONISTION
120 :
130 SOUNDCLEAR:VOLUME 15
140 ENVELOPE 1,0,0,15,0
150 :
160 WAVE 1,3,40,S   :REM SYNCHRONISATION
170 WAVE 3,1,OFF
180 :
190 OSCILLATE 3,250
200 :
210 PLAY 1,ON
220 PLAY 3,ON
230 :
240 FOR I=100 TO 3000 STEP 3
250 OSCILLATE 1,I
260 NEXT
270 :
280 SOUNDCLEAR
READY.
```

Listing 3. Ein Demonstrationsprogramm zur Synchronisation

```
100 PRINT
110 PRINT"   MOONLIGHT SHADOW  "
120 PRINT"   VON MIKE OLDFIELD  "
130 :
140 SOUNDCLEAR:VOLUME 15
150 :
160 REM NOTEN UND DAUER
170 REM FUER STIMME 1 UND 2
180 :
190 DIM N1$(200),D1(200)
200 DIM N2$(200),D2(200)
201 RESTORE
210 :
220 REM DATEN FUER STIMME 1
230 I=0
240 READ N1$(I),D1(I)
250 IFD1(I)>0THENI=I+1:GOTO240
260 :
270 REM DATEN FUER STIMME 2
280 I=0
290 READ N2$(I),D2(I)
300 IFD2(I)>0THENI=I+1:GOTO290
310 :
320 :
330 PRINT"  GESCHWINDIGKEIT (1-6) ?"
340 GET A$:IF A$=""THEN340
350 SP=VAL(A$):IFSP<1ORSP>6THEN340
360 :
370 :
380 REM HUELLKURVEN
390 ENVELOPE 1,4,8,7,10
400 ENVELOPE 2,2,11,11,11
410 :
420 S1=0:S2=0:C1=0:C2=0
430 TUNE 1,N1$(S1)
440 TUNE 2,N2$(S2)
450 WAVE 1,3,18.75
460 WAVE 2,2
470 PLAY 1,ON:PLAY 2,ON
480 :
490 :
500 REM STIMME 1
510 C1=C1+1:IFC1>(D1(S1)*SP/2)THEN PLAY
1,OFF
520 IFC1<(D1(S1)*SP)THEN 590
530 :
540 REM NAECHSTE NOTE
550 C1=0:S1=S1+1
551 IF D1(S1)<0THEN PLAY 1,OFF:PLAY 2,OF
F:END
555 PLAY 1,ON:TUNE 1,N1$(S1)
560 :
570 :
580 REM STIMME 2
590 C2=C2+1:IFC2>(D2(S2)*SP/2)THEN PLAY
2,OFF
600 IFC2<(D2(S2)*SP)THEN 640
610 :
620 REM NAECHSTE NOTE
630 C2=0:S2=S2+1:PLAY 2,ON:TUNE 2,N2$(S2
)
635 PLAY 2,ON:TUNE 2,N2$(S2)
640 GOTO510
650 :
670 :
680 REM DATEN FUER STIMME 1
690 DATA D5,3,D5,1,D5,1,E5,1,D5,1,C#5,1,
H4,4,H4,4,C#5,1,C#5,1
700 DATA D5,1,E5,3,D5,1,E5,1,F#5,2,F#5,2
,E5,1,A4,3
710 DATA D5,3,D5,1,D5,1,E5,1,D5,1,C#5,1,
H4,4,H4,4,C#5,1,C#5,1
720 DATA D5,1,E5,3,D5,1,E5,1,F#5,2,F#5,2
,E5,1,A4,3
730 DATA F#5,2,A5,2,G5,2,F#5,1,D5,1,E5,1
,F#5,2,E5,5
740 DATA F#5,3,E5,1,D5,2,D5,1,H4,1,E5,1,
F#5,2,E5,3,D5,1,E5,1
750 DATA F#5,2,A5,2,G5,2,F#5,1,D5,1,E5,1
,C#5,1,H4,1,A4,5
760 DATA D5,3,C#5,1,H4,1,C#5,1,D5,2,E5,1
0
770 DATA F#5,2,E5,6,F#5,2,E5,2,D5,1,E5,1
,F#5,2,A5,2,G5,2,F#5,1,D5,3,F#5,2,E5,6
780 DATA F#5,2,E5,6,F#5,2,E5,2,D5,1,E5,1
,F#5,2,A5,2,G5,2,F#5,1,D5,1
790 DATA E5,1,C#5,1,H4,1,A4,3,H4,1,C#5,1
800 DATA D5,3,D5,1,D5,1,E5,1,D5,1,C#5,1,
H4,4,H4,4,C#5,1,C#5,1
810 DATA D5,1,E5,3,D5,1,E5,1,F#5,2,F#5,2
,E5,1,A4,3
820 DATA D5,3,D5,1,D5,1,E5,1,D5,1,C#5,1,
H4,4,H4,4,C#5,1,C#5,1
830 DATA D5,1,E5,3,D5,1,E5,1,F#5,2,F#5,2
,E5,1,D5,7
840 DATA0,-1
850 :
860 :
870 REM DATEN FUER STIMME 2
880 DATA D3,4,D3,4,G3,4,H3,2,D3,2,A3,4,A
3,4,D3,4,A3,2,E4,1,A3,1
890 DATA D3,4,D3,4,G3,4,H3,2,D3,2,A3,4,A
3,4,D3,4,A3,2,E4,1,A3,1
900 DATA D3,4,D3,4,A3,4,A3,4,H3,4,G3,4,A
3,4,A3,4
910 DATA D3,4,D3,4,A3,4,A3,4,H3,4,G3,4,A
3,4,A3,3
920 DATA A3,1,A3,7,A3,1,A3,8,D3,4,D3,4,A
3,4,A3,3
930 DATA A3,1,A3,7,A3,1,A3,8,D3,4,D3,4,A
3,4,A3,4
940 DATA D3,4,D3,4,G3,4,H3,2,D3,2,A3,4,A
3,4,D3,4,A3,2,E4,1,A3,1
950 DATA D3,4,D3,4,G3,4,H3,2,D3,2,A3,4,A
3,4,D3,4,D3,8
960 DATA0,-1
READY.
```

Listing 4. Sogar mehrstimmige Musikstücke lassen sich mit Sound Master programmieren



Hardcopy leicht gemacht

Hardcopy-Programme konnten wir Ihnen schon viele vorstellen. Allerdings meist in Form von undurchsichtigen Maschinenroutinen. Hier zeigen wir Ihnen, wie man ein Hardcopy-Programm in Basic schreibt.

Eine Hardcopy ist die Kopie dessen, was auf dem Bildschirm zu sehen ist, auf Papier. Dazu wird Zeichen für Zeichen aus dem Bildschirmspeicher geholt und an den Drucker gesandt, sofern ein Textbildschirm kopiert werden soll. Der C 64 kennt zwei verschiedene Arten der Bildschirmdarstellung, die nicht ohne weiteres miteinander gemischt werden können: Text und hochauflösende Grafik.

Im folgenden soll die Hardcopy des Textbildschirmes näher betrachtet werden. Dazu soll ein einfaches Basic-Listing (Listing 1, Bild 1) dienen, das auf jedem Drucker den momentan am Bildschirm stehenden Text ausdruckt. Vielleicht kennen Sie das Listing in ähnlicher Form aus dem Handbuch zum Commodore MPS801. Es ist nicht ganz perfekt, aber einfach zu verstehen. Damit das Programm auch mit anderen Druckern arbeitet, müssen Sie die jeweils angegebenen Änderungen daran vornehmen.

Doch vor der Praxis etwas Theorie über die Organisation des Textbildschirmes des C 64. Der C 64 kann höchstens 1000 Zeichen am Bildschirm darstellen. Und zwar in 25 Zeilen zu je 40 Spalten. Für die 1000 Zeichen besitzt der C 64 einen reservierten Schreib-/Lese-Speicherbereich, den Bildschirmspeicher. Der Bildschirmspeicher liegt zwischen den Speicheradressen 1024 und 2023. Probieren Sie mal POKE 1024,1 aus: Ein »A« erscheint links oben am Bildschirm. POKE 1025,2 läßt ein »B« um eine Stelle nach rechts versetzt erscheinen. Mit dem Befehl PRINT PEEK(1024) erhalten Sie als Ergebnis »1«. »1« ist der Bildschirmcode des Zeichens »A«. Wenn Sie 129 anstelle von 1 in die Speicherzelle 1024 POKEn, erscheint ein reverses »A« an der ersten Bildschirmstelle. Mit den Codes 0 bis 127 können Sie alle normalen Zeichen des C 64 darstellen, mit den Codes 128 bis 255 die reversen. Diese Codes werden als »Bildschirmcodes« bezeichnet. Das folgende kleine Programm erzeugt den gesamten C 64-Zeichensatz am Bildschirm.

```
10 print chr$(147)
20 for i=0 to 255
30 poke 1024+i,i
40 next i
```

Die Bildschirmcodes dürfen nicht mit den ASCII-Codes verwechselt werden. Die ASCII-Codes gelten nämlich nur für den CHR$(x)-Befehl, mit dem Sie Zeichen über die PRINT-Anweisung drucken können. So kann das gleiche »A«, das Sie mit POKE 1024,1 auf den Bildschirm gebracht haben, auch durch PRINT CHR$(64) gedruckt werden. Nicht aber mit CHR$(1). Vergleichen Sie dazu einmal die Bildschirm- und ASCII-Tabelle im C 64-Handbuch. Sie werden feststellen, daß nicht nur die Codes verschieden sind, sondern auch, daß die ASCII-Tabelle keine Reverszeichen kennt.

Prinzip der Hardcopy

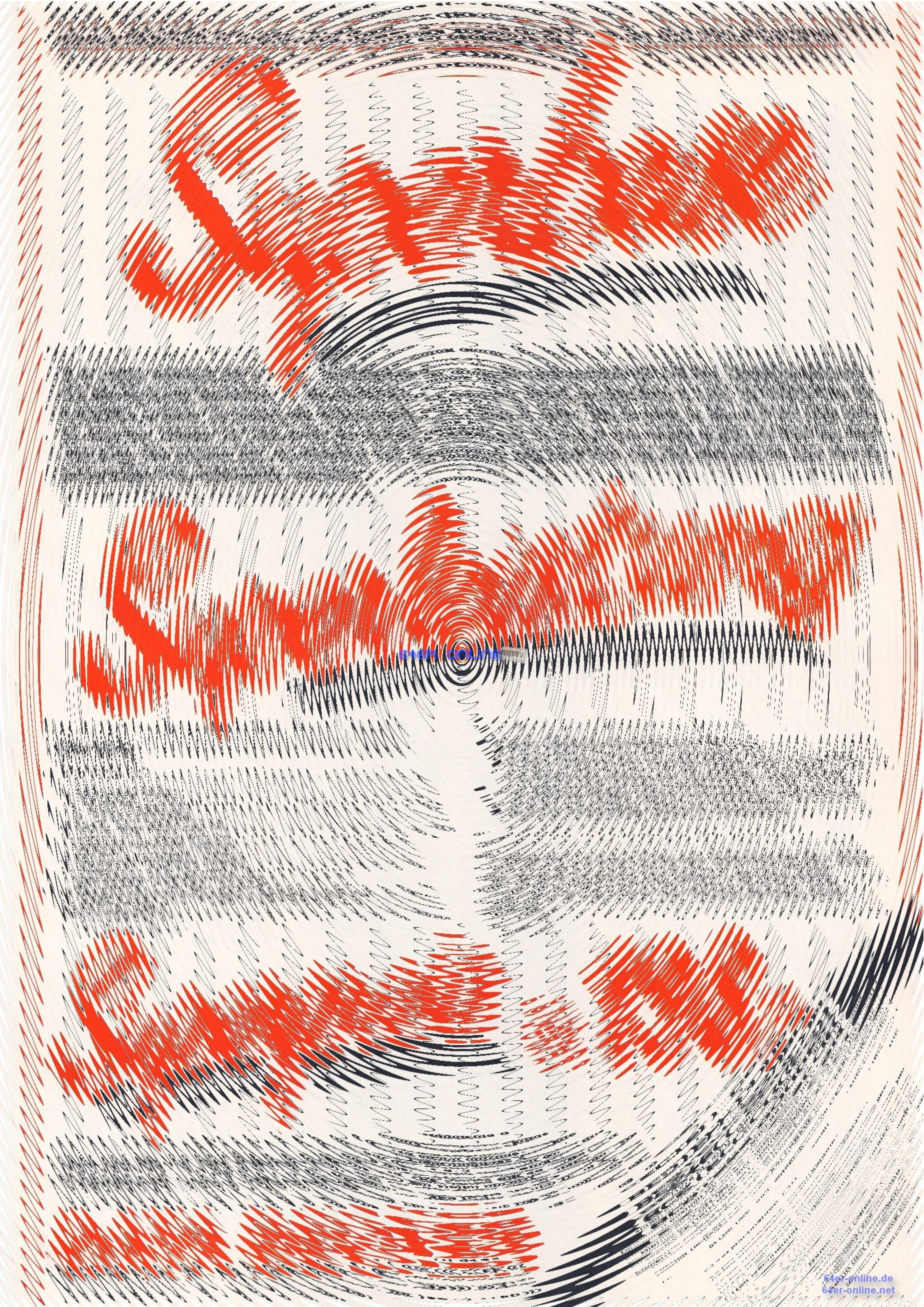
Eine Hardcopy macht im Prinzip nun nichts weiter, als ab Adresse 1024 nacheineinder alle Zeichen aus dem Bildschirmspeicher zu lesen und in der gleichen Reihenfolge an den Drucker zu schicken. Da aber der Drucker keine Bildschirmcodes versteht, müssen Sie ins ASCII-Format übersetzt werden. Die Vorschrift für eine Bildschirmcode-ASCII-Wandlung:

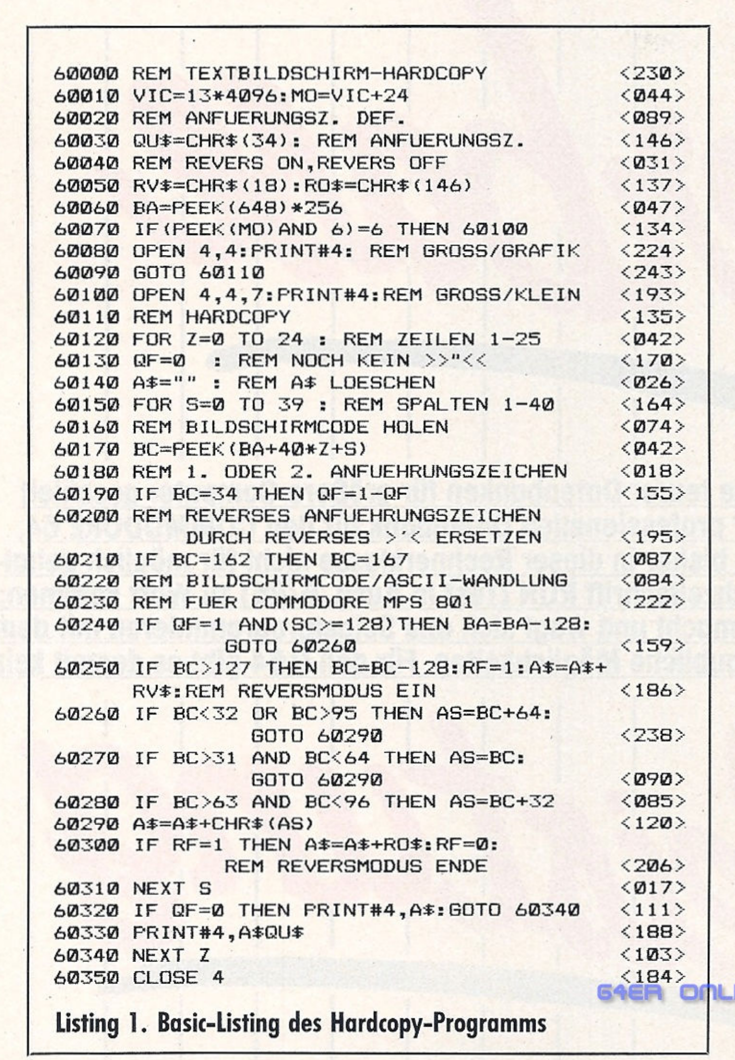
Bildschirm-code	ASCII-Korrektur-faktor
0 ... 31	+64
32 ... 63	±0
64 ... 95	+32
96 ...127	+64
128...255	1. -128

2. danach nochmalige Wandlung und Zeichen revers drucken

Um Bildschirmcode in ASCII-Code zu wandeln, müssen Sie die angegebenen Korrekturfaktoren zu den Bildschirmcodes addieren. Bildschirmcodes über 127 müssen erst um 128 verringert und dann ins ASCII-Format übersetzt werden. Aus den Bildschirmcodes der reversen Zeichen werden nämlich erst die Codes der entsprechenden »normalen« Zeichen gemacht. Die reversen Zeichen wurden normal gedruckt. Aber mit einem kleinen Trick schafft man auch die reverse Darstellung. Vorausgesetzt man hat einen Commodore MPS801 oder einen anderen Drucker, bei dem CHR$(18) und CHR$(146) den Reversdruck ein- und ausschalten. Ein reverses »A«, zum Beispiel, wird bei diesen Druckern über die Kommandofolge: PRINT CHR$(18)+"A"+CHR$(146) ausgegeben.

Das ganze Problem der Bildschirmcode-/ASCII-Wandlung kann nur mit Maschinensprache vollends gelöst oder besser umgangen werden, indem man den Textbildschirm





```
60000 REM TEXTBILDSCHIRM-HARDCOPY                  <230>
60010 VIC=13*4096:MO=VIC+24                        <044>
60020 REM ANFUERUNGSZ. DEF.                        <089>
60030 QU$=CHR$(34): REM ANFUERUNGSZ.               <146>
60040 REM REVERS ON,REVERS OFF                     <031>
60050 RV$=CHR$(18):RO$=CHR$(146)                   <137>
60060 BA=PEEK(648)*256                             <047>
60070 IF(PEEK(MO)AND 6)=6 THEN 60100               <134>
60080 OPEN 4,4:PRINT#4: REM GROSS/GRAFIK           <224>
60090 GOTO 60110                                   <243>
60100 OPEN 4,4,7:PRINT#4:REM GROSS/KLEIN           <193>
60110 REM HARDCOPY                                 <135>
60120 FOR Z=0 TO 24 : REM ZEILEN 1-25              <042>
60130 QF=0  : REM NOCH KEIN >>"<<                  <170>
60140 A$="" : REM A$ LOESCHEN                      <026>
60150 FOR S=0 TO 39 : REM SPALTEN 1-40             <164>
60160 REM BILDSCHIRMCODE HOLEN                     <074>
60170 BC=PEEK(BA+40*Z+S)                           <042>
60180 REM 1. ODER 2. ANFUEHRUNGSZEICHEN            <018>
60190 IF BC=34 THEN QF=1-QF                        <155>
60200 REM REVERSES ANFUEHRUNGSZEICHEN
        DURCH REVERSES >'< ERSETZEN                <195>
60210 IF BC=162 THEN BC=167                        <087>
60220 REM BILDSCHIRMCODE/ASCII-WANDLUNG            <084>
60230 REM FUER COMMODORE MPS 801                   <222>
60240 IF QF=1 AND(SC>=128)THEN BA=BA-128:
          GOTO 60260                               <159>
60250 IF BC>127 THEN BC=BC-128:RF=1:A$=A$+
      RV$:REM REVERSMODUS EIN                      <186>
60260 IF BC<32 OR BC>95 THEN AS=BC+64:
          GOTO 60290                               <238>
60270 IF BC>31 AND BC<64 THEN AS=BC:
          GOTO 60290                               <090>
60280 IF BC>63 AND BC<96 THEN AS=BC+32             <085>
60290 A$=A$+CHR$(AS)                               <120>
60300 IF RF=1 THEN A$=A$+RO$:RF=0:
          REM REVERSMODUS ENDE                     <206>
60310 NEXT S                                       <017>
60320 IF QF=0 THEN PRINT#4,A$:GOTO 60340           <111>
60330 PRINT#4,A$QU$                                <188>
60340 NEXT Z                                       <103>
60350 CLOSE 4                                      <184>
```

Listing 1. Basic-Listing des Hardcopy-Programms

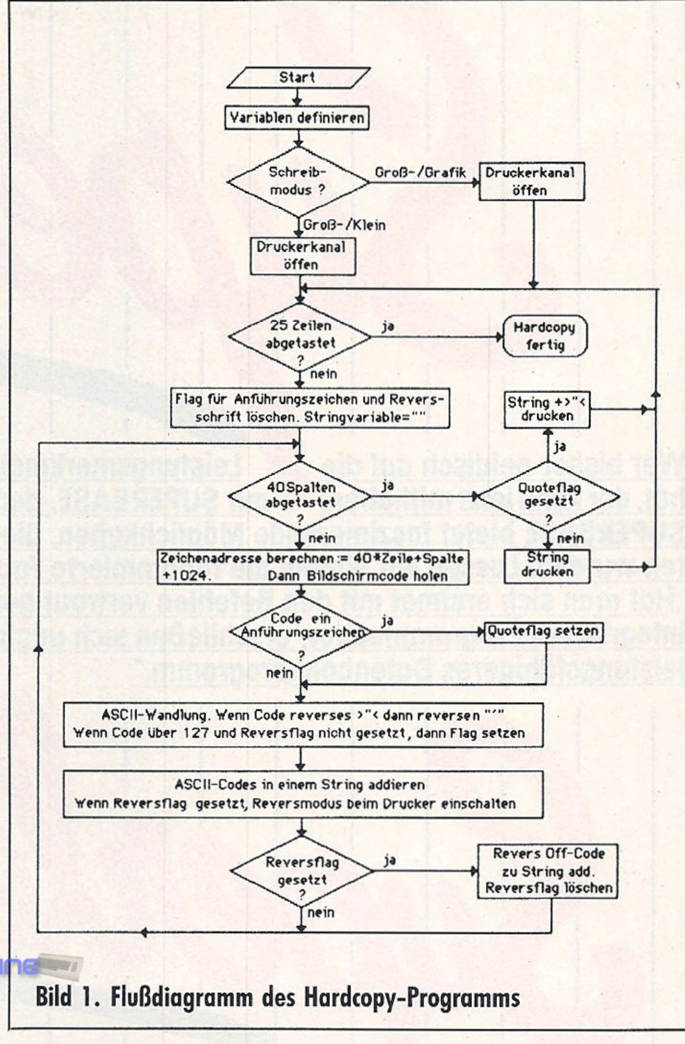


Bild 1. Flußdiagramm des Hardcopy-Programms

wie einen Grafikbildschirm behandelt. Wie das geht, erfahren Sie in der nächsten Ausgabe, in der wir Ihnen ein Maschinenprogramm vorstellen, das auf Tastendruck eine Hardcopy vom Textbildschirm druckt. Daß man nicht auf Maschinensprache verzichten kann, liegt einfach daran, daß Basic-Befehle nur auf festgelegte Speicherebenen zugreifen können.

Doch nun zurück zu der kleinen Basic-Routine, die eine Hardcopy eines Textbildschirmes druckt. In Zeile 60010 wird die Adresse des VIC-Registers 24 berechnet, das für die Schriftart verantwortlich ist: Groß-/Grafik- oder der Groß-/Kleinmodus. Die Adresse des Registers wird an die Variable MO (Modus) übergeben um mit dem PEEK-Befehl das Register auszulesen. Der Registerinhalt entscheidet in Zeile 60070 darüber, welche Sekundäradresse für den Drucker ausgewählt wird. Entweder keine Sekundäradresse für Groß/Grafik (Zeile 60080) oder »7« für Klein/Groß (60100). Ist Bit 2 und 3 von MO gesetzt, dann liegt Groß-/Kleinmodus vor.

In den Zeilen 60050 und 60060 werden einige bestimmte Steuercodes für den Drucker Variablen zugeordnet. Im Einzelnen:
QU$ = CHR$(34): ASCII-Codes eines Anführungszeichen
RV$ = CHR$(18): reverse Schrift einschalten
RO$ = CHR$(146): reverse Schrift ausschalten

Wie Sie sehen, sind die genannten Steuercodes für den Commodore Drucker MPS 801 mit denen des C 64 identisch. Haben Sie keinen MPS 801-kompatiblen Drucker, entnehmen Sie bitte die Steuercodes Ihres Druckers dem Drucker- oder Interface-Handbuch. Sollte Ihr Drucker keine Reverszeichen drukken können, bleibt Ihnen nichts weiter übrig, als reverse Zeichen in normaler Darstellung ausgeben zu lassen. Definieren Sie dazu RO$ und RV$ als CHR$(0) oder als einen anderen ASCII-Code, den Ihr Drucker ignoriert. Auf diese Weise ersparen Sie sich das Durchsuchen des Listings nach Steuercodes und deren Entfernung.

In Zeile 60050 stellt das Programm die Bildschirm-Anfangsadresse fest, die nicht grundsätzlich immer bei 1024 liegen muß. Das High-Byte dieser Adresse steht in der Speicherzelle 648, es muß mit 256 multipliziert werden. Ein Low-Byte existiert nicht, da der Bildschirm nur um ganze KByte verschoben sein kann. In Zeile 60080 wird schließlich der Druckerkanal für Groß-/Grafikmodus geöffnet, entsprechend in Zeile 60100 für Groß-/Kleinmodus. Diese Sekundäradressen müssen Sie, so wie die Steuercodes, auch auf Ihren Drucker anpassen.

Hardcopy in Basic

Ab Zeile 60110 beginnt schließlich die eigentliche Hardcopy. Die ganze Routine besteht im Prinzip aus zwei FOR-NEXT-Schleifen, die jede Zeile spaltenweise »abtasten«. Die erste Schleife (60120...60340) erhöht die Zeilenzahl (Variable Z), die zweite (60150...60310) die Spaltenzahl (Variable S). Vor Eintritt in die Spaltenschleife wird die später zu druckende Variable A$ gelöscht. Denn wenn nichts in einer Zeile steht, soll auch nichts gedruckt werden. Ebenso wird QF, die Flag-Variable für den Quote-Modus, zurückgesetzt.

Quote-Modus bedeutet, daß alle folgenden Zeichen innerhalb von

Anführungszeichen stehen. Diese Flagge wird auf »1« gesetzt, sobald ein Anführungszeichen gedruckt werden soll. Nach dem zweiten Anführungszeichen wird QF wieder auf »0« gesetzt. Innerhalb von zwei Anführungszeichen bedeutet beispielsweise CHR$(18) ein reverses »R«, sonst aber, daß alle folgenden Zeichen revers gedruckt werden. Drücken Sie mal »Cursor rechts«. Der Cursor bewegt sich nach rechts über den Bildschirm. Drücken Sie dann die »"«-Taste und dann »Cursor rechts«. Es erscheinen lauter reverse eckige Klammern, die Steuerzeichen für Cursor rechts. Sie befinden sich im Quote-Modus. Es ist sehr wichtig zwischen dem normalen und dem Quote-Modus zu unterscheiden. Auf den Code CHR$(18) hin druckt auch der Drucker alle folgenden Zeichen revers, während die Sequenz Anführungszeichen, CHR$(18) nur ein reverses »R« erzeugt, genauso wie am Bildschirm.

In Zeile 60170 werden die Zeichencodes aus dem Bildschirmspeicher mit PEEK geholt. Die Zeichenadresse berechnet sich aus der Startadresse BA+40*Zeile+Spalte. Danach wird als erstes untersucht, ob der gelesene Code ein Anführungszeichen ist. Wenn ja, wird, wie schon erwähnt, das Quoteflag QF gesetzt. Ist BC der Code für ein reverses Anführungszeichen (Bildschirmcode 162), wird BC der Code für einen reversen Apostroph zugewiesen. Diese »Umgehung« ist nötig, weil ein reverses Anführungszeichen nicht ohne weiteres gedruckt werden kann.

Bildschirmcode-/ASCII-Wandlung mit Tricks

Die Bildschirmcodes werden ab Zeile 60220 in ASCII-Format gewandelt. Eine besondere Erklärung bedarf die Zeile 60250. Da alle Bildschirmcodes größer als 127 die reversen Darstellungen der Codes 0 bis 127 sind, werden diese Codes um 128 verringert und das Reversflag RF gesetzt. Ist RF=1, dann wird vor dem eigentlichen Zeichen ein CHR$(18) (Revers ein) an den Drucker geschickt und im Anschluß an das Zeichen ein CHR$(146) (Revers aus). In Zeile 60290 werden schließlich die ASCII-Codes einer ganzen Bildschirmzeile zusammengesetzt, komplett mit allen Druckersteuerzeichen. Wurden alle Spalten einer Zeile »abgetastet« und die ASCII-Codes in A$ addiert, wird A$ an den Drucker geschickt. Der Druck einer Zeile erfolgt schließlich in 60320 oder 60330 (wenn QF gesetzt ist, also im Quotemodus gedruckt wurde). Am Ende der Hardcopy wird in Zeile 60330 der Druckerkanal geschlossen.

Diese Basic-Hardcopy-Routine können Sie eigenen Programmen anhängen und mit GOSUB 60000 aufrufen. Vergessen Sie dann aber nicht in Zeile 60340 ein RETURN einzugeben.

In den nächsten Ausgaben des 64'er Magazins setzen wir diesen Artikel fort. Da erfahren Sie, wie man in Maschinensprache eine Textbildschirm-Hardcopy-Routine programmiert. Diese Routine druckt auf Tastendruck hin eine Text-Hardcopy. Mit allen Zeichen die der C 64 darstellen kann.

Wir zeigen Ihnen dann auch, wie man eine Hardcopy des Grafikbildschirms zu Papier bringt. (hm)

Vergleichstest: Grafikerweiterungen

Grafikprogrammierung auf einem Computer ist eine sehr reizvolle Aufgabe. Gut gemachte Grafiken können nicht nur den Programmersteller, sondern auch ein Publikum, das wenig mit Computer zu schaffen hat, faszinieren.

Der Commodore 64 ist von seiner Hardware her gesehen ein äußerst leistungsfähiger Computer. Leider werden die außerordentlichen Grafik- und Soundfähigkeiten vom eingebauten Basic V2 in keiner Weise unterstützt. Will man zum Beispiel eine einfache Grafik programmieren, so steht einem eine ermüdende Programmierarbeit mit einer Unmenge von POKEs bevor. Um diesem Mißstand abzuhelfen, werden von verschiedenen Softwarefirmen Befehlserweiterungen angeboten, die eine bequeme Programmierung von Grafiken, Sprites und Sounds ermöglichen. Die Befehlserweiterungen der verschiedenen Hersteller sind in ihrem Umfang recht unterschiedlich. So gibt es Erweiterungen, die sich nur auf Grafik-, Sprite- und Soundbefehle beschränken, andere wiederum bieten eine ganze Palette von nützlichen Zusatzbefehlen wie Tools (zum Beispiel Find, Auto, Trace etc.), Befehle zur Unterstützung der strukturierten Programmierung, Befehle zum leichteren Arbeiten mit der Floppy, etc. Wir haben sieben Erweiterungen unter die Lupe genommen und werden im folgenden näher darauf eingehen, wobei von dem jeweiligen Befehlsvorrat nur die grafikspezifischen Befehle näher betrachtet werden (Tabelle 1).

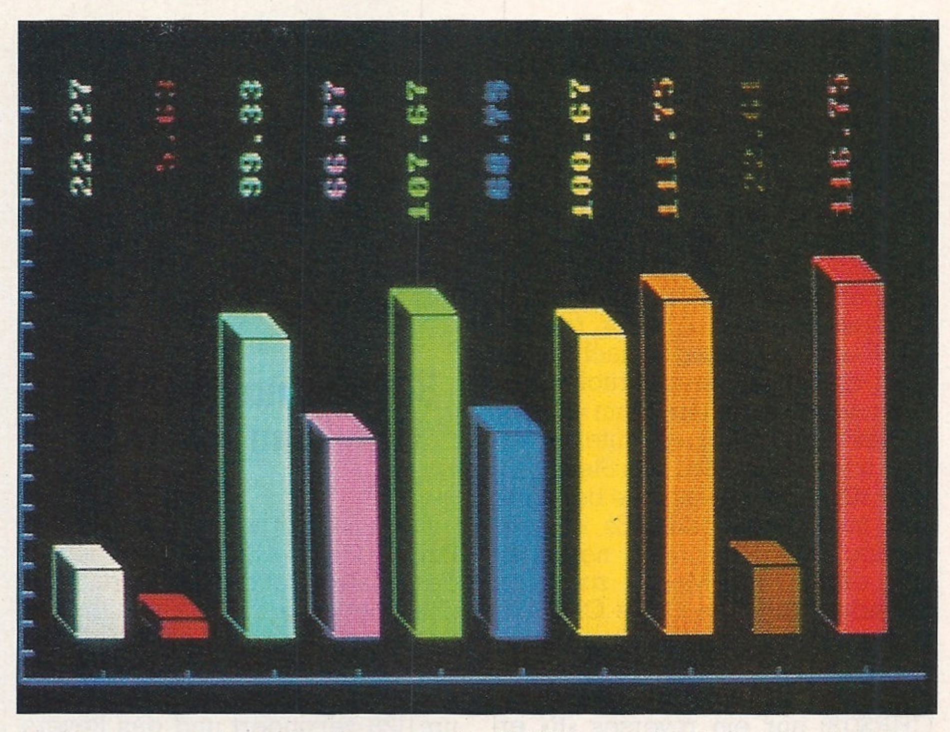
GBasic

Die Erweiterung GBasic wird als Modul geliefert, so daß sämtliche Befehle die GBasic anzubieten hat, sofort nach Einschalten des Computers zur Verfügung stehen. GBasic ist eine Erweiterung mit einem Umfang von 16 KByte. Durch Bankswitching im Modul werden dem C 64 aber nur 8 KByte Basic-Speicherplatz abgenommen. Hochauflösende Grafiken können mit GBasic auf vier voneinander unabhängigen Seiten aufgebaut werden. Der verfügbare Basic-Speicherplatz ist je nach gewählter Grafikseite unterschiedlich hoch. Bei Benutzung von Seite 0 sind nur noch 6 KByte verfügbar, bei Seite 1 noch 14 KByte und bei Seite 2 und 3 steht wieder der Gesamtspeicherplatz von 30 KByte für Basic-Programme zur Verfügung.

An Grafikbefehlen stellt GBasic die üblichen Befehle wie Plot, Line, Box, Circle, Fill etc. zur Verfügung. Mit diesem Grafikbefehlssatz ist man ganz gut bedient. Die Geschwindigkeit, mit der die Befehle ausgeführt werden, liegt im oberen Bereich der getesteten Programme. Der einzige Punkt im Test, bei dem GBasic langsam war, ist das Vollzeichnen des Bildschirmes mit 640 kleinen Kreisen (siehe Tabelle 1).

Einen interessanten Befehl besitzt GBasic, der unseres Wissen in keiner anderen Grafikerweiterung vorhanden ist. Dies ist der Befehl »Vektor«, das heißt er zieht eine Linie, die durch Anfangs- und Endkoordinaten bestimmt ist. Anders als beim Line-Befehl stoppt Vektor jedoch mit dem Ziehen der Linie, sobald er auf einen Punkt mit der gleichen Farbe trifft.

Neben den Befehlen zum Aufbau hochauflösender Grafik besitzt GBasic weiterhin Befehle zum Editieren und Steuern von Sprites. Um den Befehl zum Editieren von Sprites benutzen zu können, muß allerdings das Programm SEDIT von der mitgelieferten Diskette nachgeladen werden. GBasic unterstützt durch mehrere Befehle auch das Arbeiten mit Lichtgriffel oder Grafiktablett. Beim GBasic bekommt man eine Menge für sein Geld geliefert. Dazu zählen außer einem sehr umfangreichen Befehlsvorrat (der beschränkt ist, siehe Testbericht 64'er, Ausgabe 1/84) auch die sehr ausführliche, 116 Seiten starke Dokumentation. Das GBasic-Modul kann man als Normal- oder als Turbo-Modul erhalten. Die Bezeichnung Turbo bezieht sich auf einen eingebauten Turbolader für die Floppy.



Das Diagramm wurde mit dem Programm »Grafik 2000« erstellt. Man beachte den hochkant gestellten Text. Diese Eigenschaft haben die wenigsten Erweiterungen.

Graff

Graff ist eine Befehlserweiterung, die sich nur auf Grafik- und Spritebefehle beschränkt.

Die Erweiterung wird auf Diskette geliefert. Nach dem Laden und Initialisieren von Graff verbleiben noch 14 KByte Speicher für Basic-Programme. Graff selbst verbraucht zirka 6 KByte Speicherplatz. Sobald Graff initialisiert ist, liegt der Textbildschirm im Bereich $4000 bis $47FF. Der Zeichensatz wird ebenfalls ins RAM kopiert und zwar von $5000 bis $57FF. Weiterhin wird Speicherplatz vom 38 KByte-Basic-Speicher für Farbspeicher, HiRes-Bildschirm, Sprite-Blöcke und für einen Arbeitsbereich von Graff abgezwackt. Die Speicherkonfiguration des C 64 wird also ziemlich stark verändert. Aus diesem Grund sollte man beim Programmieren mit Graff auch sehr vorsichtig vorgehen. Da Graff bei einem Programmfehler nicht automatisch in den Textmodus zurückkehrt, kann man in diesen nur durch ein blind eingegebenes »*GF« wieder zurückkommen. Die Restore-Taste sollte man tunlichst meiden. Hat man dennoch aus Versehen einen Warmstart durchgeführt, so kommt man auch hier nur durch eine blind eingegebene Initialisierung von Graff in den Textmodus zurück.

Graff besitzt nur eine einzige HiRes-Grafik-Seite. Diese kann nur im Multicolormodus benutzt werden, das heißt man erreicht mit Graff nur eine maximale Auflösung von 160 x 200 Punkten.

Graff enthält die üblichen Grafikbefehle (siehe Tabelle 2). Alle graffspezifischen Befehle werden als Kombination von zwei Buchstaben mit vorgestelltem Sternchen eingegeben. Da der Zeichensatz im RAM steht, ist er leicht zu ändern. Mittels eines einfachen Graff-Befehls kann jederzeit wieder der ROM-Zeichensatz eingeschaltet werden.

Als Besonderheit hat Graff eine Befehlskombination, die das Erstellen von Tortengrafiken unterstützt. Außerdem können mittels einfachen Befehlen Bildausschnitte gespeichert und an eine andere Stelle des Bildschirms geschoben werden.

Die Zeiten, mit denen Graff-Befehle ausgeführt werden, sind relativ kurz. Allerdings sollte dabei in Betracht gezogen werden, daß Graff nur mit der halben möglichen HiRes-Auflösung arbeitet.

Hardwaremäßige Zusätze wie Lightpen oder Grafiktablett werden von Graff nicht unterstützt. Wenn man mit Graff arbeiten möchte, sollte man schon einen etwas größeren Einblick in die Grafik- und Spritemöglichkeiten des C 64 haben. Dann allerdings kann man auch mit Graff gute Grafiken programmieren, wie die Demoprogramme auf der Diskette beweisen. Leider ist es mit Graff nicht möglich, Hardcopies der erstellten Grafiken anzufertigen.

Die Dokumentation ist mit 16 Seiten etwas dürftig und beschränkt sich nur auf die kurze Erklärung der einzelnen Befehle. Der Text der Dokumentation ist auf der Graff-Diskette ebenfalls enthalten und kann bei Bedarf von einem Drucker ausgedruckt werden. Laden kann man diese Instruktionen allerdings nur im Normalmodus, da sie auf der Diskette etwa 127 Blocks beanspruchen, für die 14 KByte Speicherplatz von Graff also schon zu lang sind.

Supergraphik 64

Bei Supergraphik 64 handelt es sich um eine Befehlserweiterung, die Grafik-, Sprite- und Soundbefehle umfaßt. Als kleine Zugabe sind bei Supergraphik 64 noch ein paar Tool-Befehle vorhanden.

Uns lag Supergraphik 64 in seiner dritten Version vor. Geliefert wird Supergraphik 64 auf Diskette. Nach dem Laden von Supergrafik geben dem Programmierer nur zirka 10 KByte Basic-Speicherplatz verloren. Supergraphik nutzt den für Basic-Programme nicht erreichbaren Speicherplatz in der RAM-Insel und unter den ROMs.

An speziellen Grafikbefehlen hat Supergraphik 64 nicht viel zu bieten. Es fehlt sogar ein wichtiger Grafikgrundbefehl, nämlich der Befehl zum Füllen unregelmäßiger Flächen. Dies ist leider ein ziemliches Manko.

Mit Supergraphik 64 lassen sich zwei HiRes-Bildschirme und ein Textbildschirm verwalten. Grafiken können als Blockgrafik, als hochauflösende Grafik (320*200) oder als Multicolorgrafik (160*200) erstellt werden. In der Ausführungszeit der Befehle liegt Supergraphik 64 etwas unterhalb von GBasic. Nur beim Test mit dem Füllen des gesamten Bildschirms mit kleinen Kreisen liegt Supergraphik ebenso wie GBasic im mittleren Bereich.

Unterstützt wird durch Supergraphik das Zeichnen mit dem Grafiktablett. Ebenfalls möglich ist das Laden von Grafiken anderer Programme (zum Beispiel Grafiken im Koala-Format). Beim Abspeichern von Grafik kann man selbst bestimmen, in welchem Format dies geschehen soll. Die erstellten Grafiken können also auch auf andere Programme zugeschnitten und von diesen verwendet werden.

Bei Fehlern im Programm beziehungsweise am Programmende kehrt Supergraphik automatisch in den Textmodus zurück.

Die Dokumentation ist mit 81 Seiten noch sehr ausführlich und gut. Dies ist bei der Anwendung auch unbedingt nötig, da die Befehle durch eine Reihe von Parametern für die verschiedensten Zwecke (miß)braucht werden können. So ist es zum Beispiel möglich, mit dem Befehl Circle ein Rechteck oder noch extremer eine Linie zu zeichnen, je nachdem, welche Parameter wie gesetzt werden.

Als kleines Extra ist auf der Diskette ein Sprite-Editor vorhanden. Dafür sind leider keinerlei Demos auf der Diskette.

Aztek-Basic

Aztek-Basic ist keine spezielle Grafik-Erweiterung, sondern umfaßt auch alle anderen Arten von Befehlen. Uns interessiert jedoch nur der Grafikbefehlsanteil. Dieser ist mit allen für Grafikprogrammierung wichtigen Befehlen vertreten. Leider kann zur Speicheraufteilung und zur Anzahl der verfügbaren Grafikseiten nichts erwähnt werden, da uns kein Handbuch, sondern nur eine fotokopierte Liste der möglichen Befehle und ihrer Syntax vorlag. Jedoch verbleibt nach dem Laden und Starten von Aztek ein freier Basic-Speicher von rund 26 KByte Speicherbereich für den Grafikbildschirm.

Aztek-Basic führt alle Grafikbefehle nur mit sehr geringer Geschwindigkeit aus. Dies ist auch aus dem Demo von der Diskette ersichtlich. Aztek war im Test mit Abstand die langsamste Erweiterung. Gegen Aztek spricht auch, daß keine Befehle mehr abgekürzt eingegeben werden dürfen, auch die des Basic V2 nicht. Für Grafikanwendungen ist Aztek deshalb nur bedingt geeignet. Zum Programmieren von interruptgesteuerter Musik, beziehungsweise interruptgesteuerten Sprites ist Aztek dagegen zu empfehlen. Auch können interruptgesteuerte Unterprogramme realisiert werden. Einen ausführlichen Bericht finden Sie in der 64'er, Ausgabe 4/85.

Extended Graphic System

Laut Handbuch liegt das Extended Graphic System in der Tradition der Ex(tended)-Produktreihe (Exbasic Level II, Exdos, Texas-Assembler etc.). Aus diesem Grund wird auch verständlich, daß das Extended Graphic System mit Exbasic Level II zusammenarbeitet oder mittels des Exbasic Level II-Compilers compiliert werden kann. Dies ist bei Befehlserweiterungen für den C 64 nicht so ohne weiteres selbstverständlich.

Nach dem Einladen von der Diskette verbleiben für Basic-Programme etwas mehr als 15 KByte Speicherplatz. Dabei verbraucht aber ein gleichzeitig im Computer vorhandenes Exbasic Level II keinen weiteren Speicherplatz, das heißt das Extended Graphic System läßt den Modulspeicherplatz unberührt.

Das Extended Graphic System umfaßt alle wesentlichen Grundbefehle zur Grafikprogrammierung. Leider fehlt bei dieser Spracherweiterung der Befehl zum Füllen von Flächen. Dies schränkt die Verwendbarkeit etwas ein. Das ist schade, denn durch seine Geschwindigkeit, die im mittleren Bereich liegt (außer dem Circle-Befehl, der hier sehr schnell arbeitet) ist Extended Graphic System eine recht passable Erweiterung, auch im Hinblick auf Compilierbarkeit und Zusammenarbeit mit anderen Ex-Erweiterungen.

Zum Extended Graphic System gehört nicht allein die Befehlserweiterung, sondern auch darauf abgestimmte Zeichen-, Sprite- und Grafikeditoren. Diese drei werden ebenfalls auf der Diskette mitgeliefert. Das Extended Graphic System kann zwei Grafikseiten verwalten. Allerdings muß bei Grafikseite 0 mit Vorsicht hantiert werden, da hier auch der Zeichensatz steht. Im Extended Graphic System kann nämlich auch der Zeichensatz beliebig abgeändert und auf Diskette abgespeichert werden. Sollte einem trotzdem das Mißgeschick passieren und der Zeichensatz gelöscht worden sein, so muß der Standardzeichensatz oder ein selbst erstellter im Blindverfahren von Diskette nachgeladen werden.

Vorteilhaft bei Extended Graphic System ist auch das frei definierbare Format beim Abspeichern. Beim Einladen von Fremdgrafiken müssen diese jedoch mittels kurzer Umwandlungsprogramme an das Extended Graphic System angepaßt werden. Dies heißt unter anderem auch, daß Grafikbilder, die mit Extended Graphic System erstellt worden sind, in anderen Programmen benutzt werden können. Extended Graphic System war mit den Ausführungszeiten der Befehle teilweise schnell, teilweise langsam. So wurde der Befehl Circle beim Füllen des Bildschirms mit kleinen Kreisen sehr schnell ausgeführt, bei den konzentrischen Kreisen dagegen

war Extended Graphic System langsam.

Die Dokumentation ist mit 70 Seiten im DIN-A4-Format sehr umfangreich (wie bei allen Interface-Age Produkten) und geht auch auf die tieferen Zusammenhänge beim Programmieren von Grafiken ein. Das Handbuch wird auch nicht als reine Programmdokumentation vorgestellt, sondern als Kursus oder Lehrbuch für Grafikprogrammierer ohne besondere Vorkenntnisse. Eine Fortsetzung dieses Kurses soll in nächster Zeit folgen.

Macro-Basic

Ein völlig anderer Weg, das Commodore Basic um neue Befehl zu erweitern, wird von Macro-Basic beschritten. Aus einer Sammlung kleiner Erweiterungsmodulen kann man sich mittels eines Linkprogrammes jede mögliche Kombination von Befehlen zusammenstellen (siehe auch Testbericht 64'er, Ausgabe 6/85). Man kann also die nur für ein Programm zusätzlich benötigten Befehle zu einer Befehlserweiterung zusammenfassen. Zur Zeit sind in Macro-Basic etwa 160 Befehle verfügbar. Alle Befehlsmodule (außer dem Grafikmodul) sind kompatibel zu den anderen Commodore-Computern. Jedes Macro-Basic-Modul stellt einen Befehl dar, nur das Grafik-Modul fällt hier etwas aus der Rolle. Im Grafik-Modul sind alle Befehle zu einem großen Modul zusammengefaßt. Da beim Programmieren von Grafiken in der Regel sowieso fast alle Grafikbefehle gebraucht werden, ist dies nicht weiter tragisch. Außerdem ist das Grafik-Modul nur auf dem C 64 lauffähig.

Es umfaßt insgesamt 23 Befehle. Mit diesen Befehlen kann man gut Grafiken programmieren, da alle gängigen Funktionen vorhanden sind.

Die Geschwindigkeit, mit der Macro-Basic-Befehle ausgeführt werden, ist annehmbar. Mit den meisten Befehlsausführungszeiten lag Macro-Basic im Mittelfeld. Der Befehl zum Füllen des Bildschirms mit kleinen Kreisen wurde jedoch enorm schnell ausgeführt. Auch das Zeichnen von konzentrischen Kreisen beziehungsweise eines Kreises allein wird sehr schnell erledigt.

Die Dokumentation ist aufgrund der vielen Befehle sehr umfangreich. Generell wird pro Befehl etwa eine DIN-A4-Seite zur Erklärung gebraucht. Den Grafikbefehlen wird weniger Platz gewidmet. Auf einer Seite werden zwei bis drei Befehle erläutert. Trotzdem reicht auch hier der Platz noch für kurze Beispielprogramme.

Simons Basic

Die wohl älteste und bekannteste Befehlserweiterung dürfte Simons Basic sein. Simons Basic ist auf Modul und auf Diskette erhältlich und zählt zu den Programmen, die außer Grafik auch eine Reihe anderer Funktionen enthalten.

Simons Basic kann einen HiRes-Bildschirm verwalten und zwar sowohl in Normal- wie auch in Multicolor. Befehle zur Unterstützung von Lichtgriffel und Grafiktablett sind ebenfalls integriert.

In der Geschwindigkeit tut sich Simons Basic nicht besonders hervor. Es zählt eher zu den langsamen Programmen. Positiv an Simons Basic dürfte die weite Verbreitung sein. Deshalb werden auch ab und zu in Zeitschriften entsprechende Listings zu finden sein. Das Handbuch von Simons Basic mit über 70 Seiten ist zur Einführung gut geeignet. Es wird aber nur jeweils die genaue Syntax der Befehle erklärt. Für einen erfahrenen Programmierer ist dies kein Handicap. Ein Anfänger in der Grafikprogrammierung sollte auf alle Fälle ein entsprechendes Handbuch zu Rate ziehen. Gerade für Simons Basic sind gute Bücher zu erhalten.

Grafik 2000

Beim Test wurde auch Grafikerweiterung mitgetestet, die als Leser-Listing im vierten 64'er-Sonderheft Grafik/Drucker veröffentlicht wurde. Dieses Programm kann im Umfang der integrierten Befehle durchaus mit den anderen Erweiterungen mithalten. Zur Verfügung stehen 41 neue Befehle und zwei neue Test-Funktionen. Auch die Geschwindigkeit, mit der die Befehle ausgeführt werden, kann als hervorragend bezeichnet werden.

Grafik 2000 verwaltet zwei Grafikschirme. Die Erweiterung selbst und die Grafikschirme liegen außerhalb des Basic-Speichers, so daß dem Benutzer weiterhin der gesamte Basic-Speicherplatz von 38911 Byte zur Verfügung steht.

Bei Grafik 2000 ist man gegen grobe Fehleingaben sehr gut geschützt. Das Programm kann nur durch unsinnige POKEs zum Absturz gebracht werden. Alle Befehle lassen sich wie beim Basic V2 abkürzen, weil sie als Tokens gespeichert werden; selbst nach einem THEN ist kein Doppelpunkt erforderlich. Als besonderer Befehl kann bei Grafik 2000 der Befehl »Window« genannt werden. Er ermöglicht die Darstellung eines Grafikfensters im Textmodus. Dies ist sehr nützlich, wenn man zum Beispiel ein Adventure programmieren möchte. Rahmen- und Hintergrundfarben können wegen der eigenen NMI/Break-Routine ohne Unterbrechung des laufenden Programms durch einen Druck auf die Restore-Taste gesetzt werden.

Testbedingungen

Alle Programme wurden zwecks Ermittlung der Geschwindigkeit einem kurzen Test unterzogen. Hierbei mußte als erster Test das Vollschreiben des Bildschirmes mit 640 kleinen Kreisen und kleinen Quadraten vollzogen werden (Bild 1 und 2). Beim Zeichnen der kleinen Kreise wurden die größten Unterschiede festgestellt. Simons Basic und Aztec-Basic waren mit Abstand die langsamsten Programme. Spitzenreiter auf der anderen Seite ist Macro-Basic (fast 70mal schneller!).

Beim zweiten Test, dem Zeichnen von zehn konzentrischen Kreisen beziehungsweise Rechtecken, waren die Unterschiede nicht ganz so gravierend (siehe Bild 3 und 4). Allerdings war auch hier Aztec beziehungsweise Simons Basic wieder das langsamste Programm. Macro-Basic behielt seine Rolle als Schnellläufer, wurde jedoch geschlagen von Grafik 2000.

Als letzter Test war das Füllen je eines Kreises und eines Quadrats angesagt. Auch hier waren die Unterschiede nicht so extrem (Bild 5 und 6). Als einziges Produkt war Graff hier sehr schnell, was vielleicht begründet ist in der geringen Auflösung von 160 x 200 Punkten, die Graff hat.

Sie werden bemerken, daß einige Werte nicht ermittelt wurden, zum Beispiel das Zeichnen von Rechtecken (spezieller BOX-Befehl fehlt) bei Grafik 2000 oder das Füllen von Flächen, bei Supergraphik und beim Extended Graphic System (kein FILL-Befehl). Diese Funktionen könnten zwar simuliert werden, würden dann jedoch den gesamten Zeitrahmen sprengen.

Zusammenfassung:

Alle getesteten Programme sind auf ihre Weise weniger bis sehr gut zum Programmieren von Grafik einzusetzen. Will man nur ab und zu kleinere Grafiken erstellen, so ist



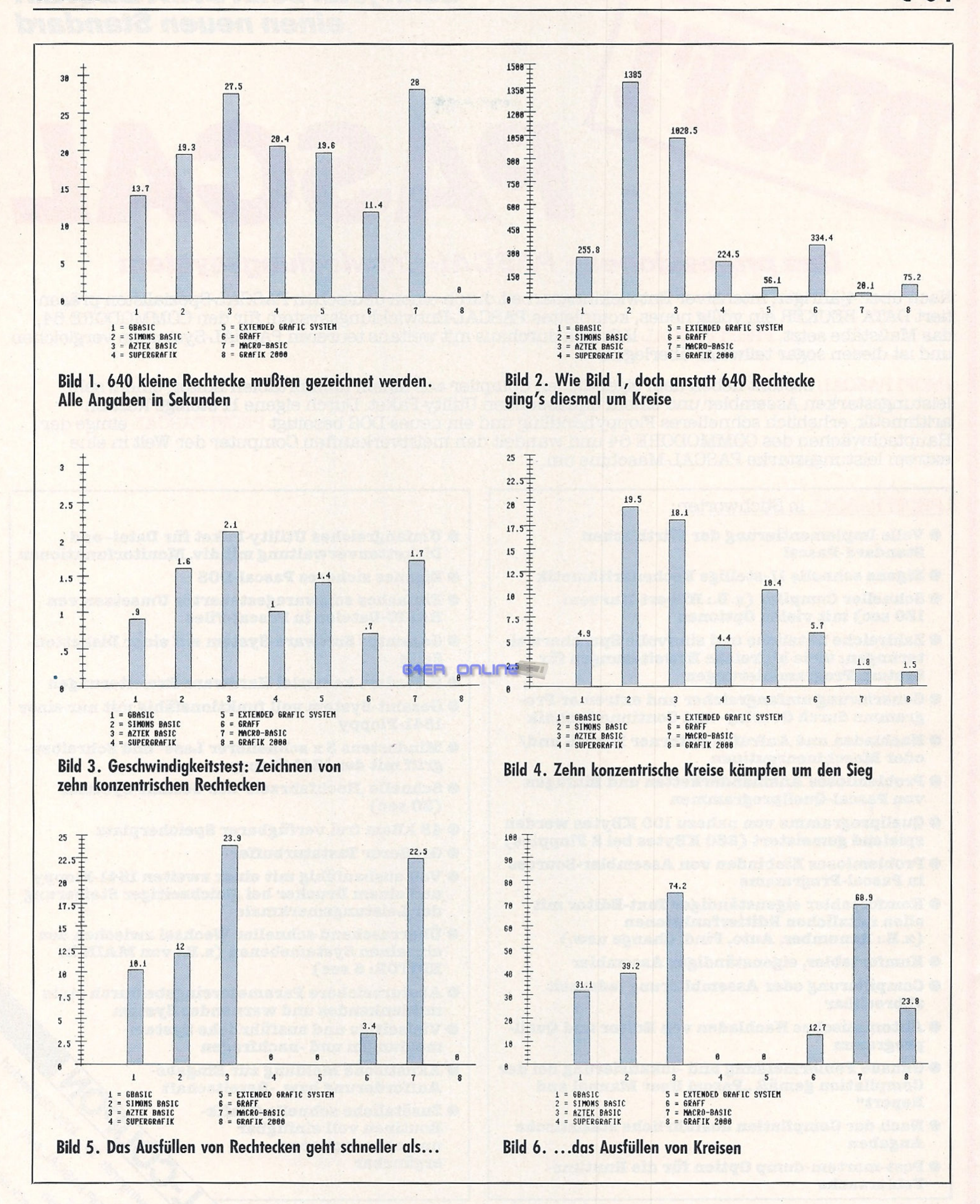


Bild 1. 640 kleine Rechtecke mußten gezeichnet werden. Alle Angaben in Sekunden

Bild 2. Wie Bild 1, doch anstatt 640 Rechtecke ging's diesmal um Kreise

Bild 3. Geschwindigkeitstest: Zeichnen von zehn konzentrischen Rechtecken

Bild 4. Zehn konzentrische Kreise kämpfen um den Sieg

Bild 5. Das Ausfüllen von Rechtecken geht schneller als...

Bild 6. ...das Ausfüllen von Kreisen

man mit reinen Grafikprogrammen wie Graff eigentlich schon sehr gut bedient. Reine Grafik-Erweiterungen kann man auch dann ins Auge fassen, wenn man schon über geeignete Toolkits etc. verfügt. Hier wären Graff, Supergraphik 64 oder das Extended Graphic System zu nennen. Bei den zwei letztgenannten ist es eigentlich schade, daß sie nicht über den Fill-Befehl verfügen. Der Minuspunkt von Graff dürfte die Beschränkung auf reine Multicolorgrafiken sein.

Sollte man noch nicht im Besitz zusätzlicher Programmierhilfen sein oder auch Befehlen, die die strukturierte Programmierung unterstützen, so kommt man beim Kauf von GBasic, Simons Basic, Aztec-Basic oder Macro-Basic sicherlich preis-

lich günstiger weg. Spitzenreiter dürfte hierbei das GBasic-Modul sein, da eine große Menge an Befehlen sofort nach Einschalten des Computers verfügbar sind. Macro-Basic kann in dieser Hinsicht natürlich mit GBasic nicht verglichen werden, da die Anzahl der Befehle bei Macro-Basic wesentlich umfangreicher ist. Macro-Basic ist wegen der Möglichkeit der individuellen Befehlszusammenstellung sehr gut geeignet, kompatible Programme für andere mit Macro-Basic ausgestattete Commodore-Computer zu schreiben. Dies ist vor allem dann von Vorteil, wenn man seine Programme vermarkten möchte.

Aztec-Basic kann für Grafikprogrammierung nicht uneingeschränkt empfohlen werden. Dies vor allem wegen der langsamen Zeichengeschwindigkeit.

Simons-Basic wie auch Aztec-Basic ist für alle die interessant, denen es nichts ausmacht eine Befehlserweiterung vor der Benutzung einzuladen und zu initialisieren. Gegen Simons wie auch Aztec-Basic spricht jedoch die nicht gerade überragende Geschwindigkeit, wobei Simons Basic den Vorteil der weiten Verbreitung hat.

(Hermann Zwartscholten/gk)

Info: GBasic: Omikron, Erlachweg 15, 7534 Birkenfeld
Simons Basic: Commodore, Lyonerstr. 38, 6000 Frankfurt 71
Aztek-Basic: Aztek Software, Auf der Heide 18, 3253 Oldendorf
Supergraphik 64: Data Becker, Merowingerstr. 30, 4000 Düsseldorf 1
Exteded Graphic System: Interface Age, Josephsburgerstr. 6, 8000 München 90
Graff: Rushware, An der Gümpgebrücke 15, 4044 Kaarst 2
Macro Basic: S&A, Langgasse 93, 5216 Mondorf
Grafik 2000: 64'er, Sonderheft 4 (Grafik/Drucker), Markt & Technik, Hans-Pinsel-Str. 2, 8013 Haar bei München

Wirkung des Befehls	GBasic	Aztek-Basic	Supergraphik 64	Extended Graphic System 1.)	Macro Basic 2.)	Graff	Simons Basic	Grafik 2000	Commodore 128
Punkt zeichnen	Plot	PSET	Plot	DOT	GDOT	*P.P	PLOT	SPOINT, CPOINT	DRAW
Linie zeichnen	Line	Line	Plot TO	DRAW	GDRAW, GLINE, GSLINE	* PL	LINE	SLINE, CLINE	DRAW
Rechteck zeichnen	BOX	BOX	Frame	Frame	GFRAME	* PB	REC	—	BOX
gefülltes Rechteck zeichnen	Block	GRAPH B	Fill	—	GBLOCK	* FB	BLOCK	CCIRCLE	—
Radius zeichnen	mit Vektor	ANGL.	Plot to	—	—	* PR	ANGL	—	—
Kreis zeichnen	Circle	GRAPH C	Circle	Circle	GCRL	* PC	CIRCLE	SCIRCLE, CCIRCLE	CIRCLE
Kreisbogen zeichnen	Circle	—	Circle	—	—	* PC	ARC	—	DRAW
Beliebige Figur zeichnen	—	DRAW	DRAW	—	GMACRO	* DP	DRAW	—	DRAW
Lage eines Punktes bestimmen	Plot + Mode	—	wird über Parameter bestimmt	—	GTEST	—	TEST	TEST	RDOT
Fläche füllen	Fill	GRAPH F	—	—	GFILL	* FA/* FF	PAINT	FILL	PAINT
eine Figur drehen	—	—	SCALE =	—	—	—	ROT	SCROLL	—
letzte Grafik zurückholen	HGR	HiRes x	GMODE	Bitmap n	Grafik on	* GR	HIRES	Mode	GRAPHIC
Grafikbildschirm einschalten	HGR	HiRes x	GMODE	HIRES ON	GRAFIK ON	* GR	HIRES	Mode	GRAPHIC
Modus wählen (Multicolor Normalgrafik)	HGR	GRAPH M	GMODE	—	GMULTI, GHIRES	—	MULTI	—	GRAPHIC, WIDTH
Grafik ausschalten	TEXT	HiRes x	geschieht automatisch	HIRES OFF	GRAFIK OFF	* GF	HIRES/automatisch	Mode	GRAPHIC
Zeichenfarbe wählen	Color a,b,c,d	INK, BORDER, PAPER	Color, SCOL, PCOL	PENCOLOR	über Befehlsparameter	* TC, * GC	über Parameter	Color	Color
Grafikschirm löschen	Clear n	CLG	GClear	GClear	GCLEAR	* CL	—	Clear	SCNCLR
Punkte setzen, löschen, invertieren (Modus)	Mode x	—	GMODE	jeweils über Parameter	GMODE	über Parameter	über Parameter	über Parameter	über Parameter
Texte in Grafik schreiben	Print	GRAPH P	TEXT	WRITE	GTEXT	* SP	CHAR TEXT	STEXT	CHAR
Größe der Textbuchstaben einstellen	SIZE	—	—	—	—	über Parameter	—	über Parameter	über Parameter
Verknüpfen von mehreren Grafiken	ADD	—	GComb	—	—	* XA	—	COMB	—
Grafik ausdrucken	HCopy	—	HCopy	USR(x)	GHCOPY	—	HRDCOPY	—	—
Zusätzliche Befehle	Vector	—	—	—	—	* PI (Piecechart), * BP, * FP, *GP	—	HMARK, VMARK, DUPLICATE, WINDOW	SCALE, LOCATE
Befehle für Lichtgriffel	PEN	—	PEN	—	—	—	PEN	—	PEN

1.) alle Befehle mit jeweils vorangestelltem Call
2.) alle Befehle mit jeweils vorangestelltem &

Die wichtigsten Grafik-Befehle der getesteten Basic-Erweiterungen. Viele der Befehle besitzen durch Verändern der (hier nicht angegebenen) Parameter noch umfangreichere Möglichkeiten.

Versteht Sie Ihr Computer?

Wissen Sie, wie eine Tastatur oder eine Maus funktioniert? Wir geben Ihnen darauf die Antwort. Außerdem sagen wir Ihnen, welche Einsatzgebiete die verschiedenen Eingabegeräte haben.

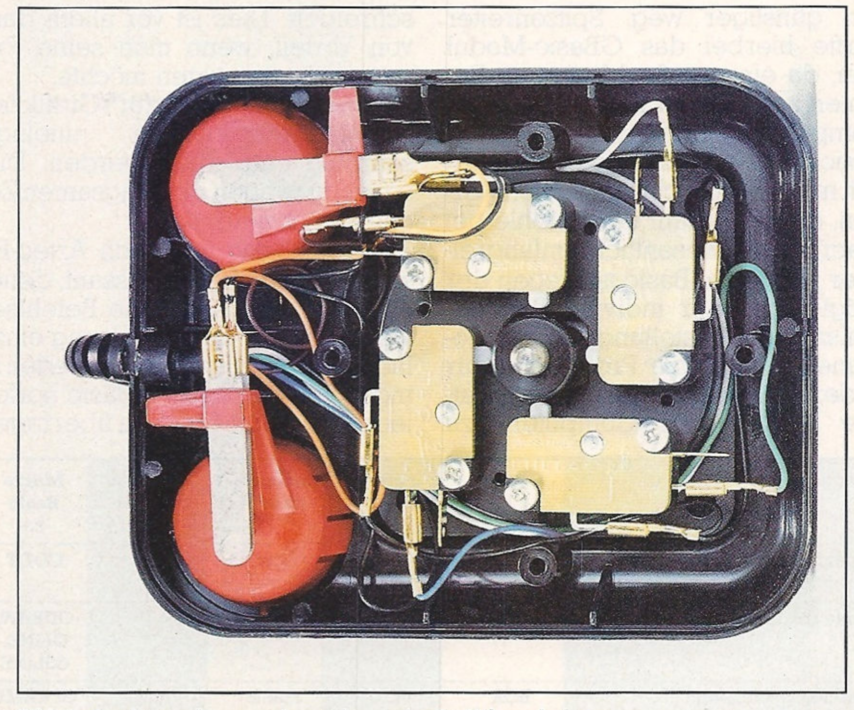


Bild 4. Qualitativ bessere Joysticks sind mit Mikroschaltern ausgestattet

Eine Tastatur finden Sie bei jedem Computer. Egal, wie alt dieser ist, oder von welcher Firma er gebaut wird. Der Grund dafür ist prinzipiell schnell erklärt: Die Tastatur ist einfach am vielseitigsten in ihrer Verwendung. Sie können Programme in den Computer hacken oder »Pacmänner« füttern.

Die Vielseitigkeit einer Tastatur ergibt sich aus der Zahl der Funktionen, die über einzelne Tasten gesteuert werden. Eine Tastatur ist prinzipiell nämlich nichts weiter als ein Auswahlmenü. Ein sehr großes allerdings. Beim C 64 hat das »Menü« 64 Tasten, die mit bis zu vier verschiedenen Funktionen oder Zeichen belegt sind.

Eine Tastatur stellt also ein wesentlich komplexeres Eingabegerät dar, als beispielsweise ein simpler Joystick mit seinen acht Richtungen. Man kann sogar sagen, es ist das komplexeste Gerät, mit dem primär Daten eingegeben werden können. Primär soll hier heißen, daß die Daten mit der Hand eingegeben werden und nicht über Lochstreifen oder ähnlichem.

Die RESTORE-Taste soll hier nicht berücksichtigt werden. Denn ihre Funktion, das Auslösen eines NMIs, hat mit der normalen Zeicheneingabe überhaupt nichts gemein. Schon eher etwas mit einer Notbremse. Denn ein NMI (Non Maskable Interrupt, unbedingter Interrupt) fordert den Prozessor auf, seine momentane Arbeit zu unterbrechen und an eine bestimmte Adresse zu springen.

Stellen Sie sich vor, Sie sollten mit einem Joystick Buchstaben aus einem Bildschirmmenü auswählen und damit einen Brief zusammensetzen. Eine Qual. Denn die Auswahl der Zeichen über ein Bildschirmmenü wäre einfach unerträglich langsam. Im Gegensatz zu einer Tastatur, bei der Sie mit Ihrem Finger sofort die richtige Taste drücken; ganz im Gegensatz zu einem Bildschirmmenü, in dem Sie erst lange mit dem Joystick umherirren, bis Sie den richtigen Buchstaben anvisiert haben.

Wegen der Vielseitigkeit und Geschwindigkeit der Dateneingabe ist eine Tastatur (noch) unersetzbar. »Noch« deshalb, weil die momentanen Spracherkennungsprogramme nicht leistungsfähig genug sind. Ist die Technik aber einmal soweit, daß ein Computer Sie akustisch verstehen kann, werden Tastaturen vielleicht bald in den Hintergrund treten. Akustisch verstehen soll in diesem Zusammenhang »uneingeschränkt verstehen« bedeuten. Ein Brief wird dann nicht mehr getippt, sondern gleich dem Computer diktiert werden. Die Tastatur wird dann wahrscheinlich nur noch zur Korrektur der Spracheingaben gebraucht. Folientastaturen kennen Sie be-

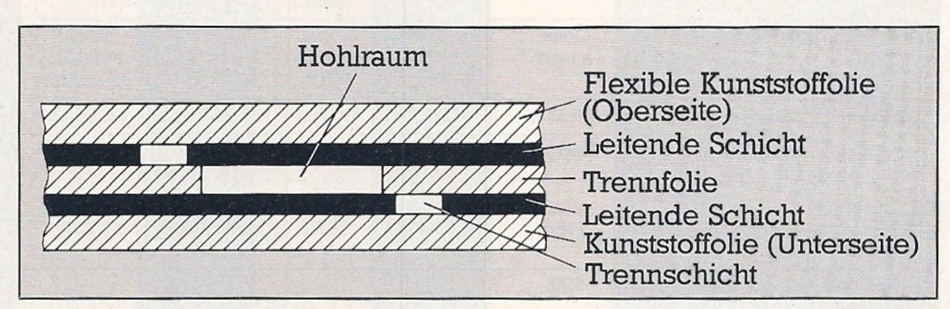


Bild 1. Der Querschnitt einer Folientastatur

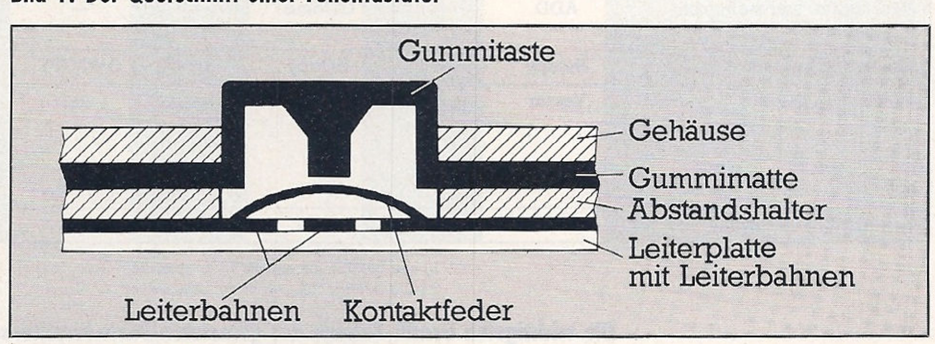


Bild 2. Nicht viel besser als eine Folientastatur: die Gummitastatur

stimmt! Falls Sie noch nicht mit einem Atari 400 oder einem Sinclair ZX81 Bekanntschaft geschlossen haben, haben Sie sicherlich diese Tastaturen schon bei einem Taschenrechner »genossen«. Zum Glück findet diese Spezies inzwischen bei Computern keine Verwendung mehr. Mit steigender Beliebtheit werden sie jedoch bei Taschenrechnern eingesetzt. Denn waren vor einigen Jahren noch Leistungsmerkmale die Unterscheidungskriterien für einen Kauf, scheint es heute die Größe zu sein; insbesondere die Bauhöhe. Bild 1 zeigt Ihnen den Querschnitt einer Folientastatur. Sie sehen drei Schichten von Plastikfolien. Die obere besteht aus sehr elastischem Material. Darunter befinden sich zwei leitfähige Kunststoffschichten, die durch eine gelochte Isolationsfolie getrennt sind. Drückt man nun eine Taste, werden die beiden leitfähigen Schichten miteinander verbunden. Und zwar an der Stelle, an der die Isolationsfolie ein Loch hat. Wird die Taste losgelassen, federn die Folien in ihre Ruheposition zurück und der Kontakt wird unterbrochen. Die Herstellung dieser Tastaturen ist sehr billig.

Die Billiglösung — Folientastaturen

Denn man benötigt keine besonderen mechanischen Teile. Ein anderer Grund für den niedrigen Preis liegt darin, daß die Einzeltasten nicht auf einer Platine zusammengestellt werden müssen. Die Grundfolie ist ja auch gleichzeitig die Trägerplatte.

Die Gummitastatur

Ähnlich der schon beschriebenen Folientastatur funktioniert die »Gummitastatur«. Hier ist die Tastaturoberfläche keine Folie mehr, sondern ein Feld aus Gummitasten, die aber den Namen »Noppen« mehr verdienen als Tasten (Bild 2). Die Gummitasten stellen eigentlich nur eine Verlängerung des Kontaktpunktes der Folie nach oben hin dar. Einen Druckpunkt schaffte man durch Einbau einer Kontaktfeder. Das ist ein rundes, konvex gewölbtes Metallplättchen, das über die Gummitaste federnd niedergedrückt wird. Dabei werden zwei Kontakte auf der Grundplatte überbrückt. Gegenüber der Folienversion hat die Gummitastatur wegen dem »Knackfrosch« den enormen Vorteil eines Druckpunktes. Denn

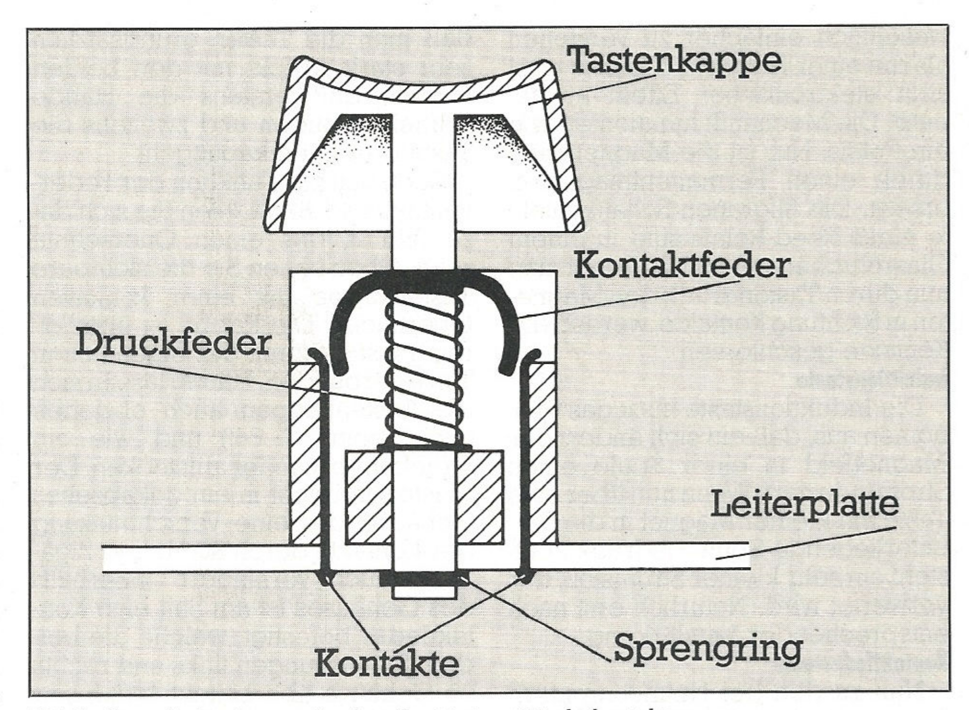


Bild 3. Querschnitt einer professionellen Taste mit Federkontakten

erst ein Druckpunkt verschafft ein sicheres Schreibgefühl. Bei einer Folientastatur weiß man nie genau, ob man eine Taste stark genug gedrückt hat oder nicht. Eine Erfolgsbestätigung erhält man erst, wenn eine Reaktion am Bildschirm zu sehen ist. Aber dennoch ist die Gummitastatur noch nicht der Weisheit letzter Schluß. Denn die Gummikappen sind meist recht wackelig und klein. Die fehlende Stabilität ergibt sich einerseits aus der mangelhaften Führung der Gummikappe und andererseits aus den Eigenschaften von Gummi selbst. Die mangelde Tastenführung merkt man daran, daß sich die Tasten sehr leicht verkanten lassen. Außerdem ist die Tastenform meist alles andere als ergonomisch. Entweder sind die Oberflächen völlig gerade oder konvex (nach oben gewölbt, was keine Griffsicherheit zuläßt). Der Tippfinger »zentriert« sich deshalb nicht beim Drücken einer Taste auf der Tastenoberfläche, was unweigerlich zu einem Verkanten führt. Außerdem sind alle Folien- und Gummitastaturen völlig plan. Im Gegensatz zu jeder professionellen Tastatur, die nach hinten ansteigt. Meist mit wachsender Steigung.

Die Guten

Aber nun Schluß mit den Einfachtastaturen. Die vier professionellen Lösungen sind gefragt:

1. Hall
2. Reed
3. Induktion
4. Feder-Kontakt

Die ersten drei sind bei Heimcomputern noch nicht zu finden. Sie sind einfach noch zu teuer.

Eine Hall-Tastatur funktioniert über einen »elektronischen Kontakt«. Wie der Name schon sagt, wird hier der Halleffekt ausgenutzt.

Der Halleffekt

Setzt man einen stromdurchflossenen Leiter einem Magnetfeld aus, tritt an den zur Stromrichtung parallelen Seiten ein Potentialunterschied auf. Stellen Sie sich dazu am besten ein Blech vor, durch das ein kleiner Strom fließt. Stromrichtung gleich Längsrichtung des Bleches. Schließt man an den Längsseiten an genau gegenüberliegenden Punkten ein Spannungsmeßgerät an, erhält man keinen Ausschlag. Das Meßgerät spricht erst dann an, wenn Sie senkrecht zum Blech ein Magnetfeld anlegen. Bildlich gesehen, indem Sie das Blech in einen Hufeisenmagneten schieben. Sie werden auch feststellen, daß die Stromrichtung von der Magnetfeldrichtung abhängt. Dieses Phänomen tritt immer auf, wenn Sie einen Leiter in ein Magnetfeld bringen. Der Effekt beruht darauf, daß geradlinig bewegte Elektronen in einem senkrecht zur Bewegungsrichtung stehenden Magnetfeld eine Kreisbahn einschlagen wollen. In dem erwähnten Blech also zur Seite fließen und einen meßbaren Potentialunterschied verursachen.

Reed-Taste

Eine Reed-Taste funktioniert ebenfalls magnetisch. Vielleicht haben Sie sogar schon selbst mit Reed-Relais zu tun gehabt. Die Funktion ist

wesentlich einfacher zu verstehen als die einer Halltaste, denn es wird kein elektronischer Effekt ausgenutzt. Die Mechanik funktioniert wie ein Relais. Nur ist die Magnetspule durch einen Permanentmagneten ersetzt. Die filigranen Schaltkontakte eines Reed-Relais sind in einem Glasröhrchen gekapselt. Bringt man nun durch Tastendruck den Magneten in Richtung Kontakte, werden die Kontakte geschlossen.

Induktionstaste

Die Induktionstaste nutzt das Phänomen aus, daß ein sich änderndes Magnetfeld in einer Spule einen Strom induziert. Wenn nun über eine Taste ein kleiner Magnet in die darunterliegende Spule gedrückt, entsteht ein sehr kleiner Stromstoß, der verwertet wird. Natürlich erst nach entsprechender Verstärkung.

Kontaktfedertaste

Nun zu den bei Heimcomputern gebräuchlichen Tasten — den Federtasten. Diese Tasten funktionieren wie ein gewöhnlicher Taster. Sie weisen aber eine kompliziertere Mechanik auf (Bild 3). Der Grund dafür ist, daß ein leichter Druck genügen muß, um die Finger nicht zu ermüden. Außerdem muß noch ein spürbarer Druckpunkt, ein hörbares »Klicken« oder wenigstens ein leicht erreichbarer Endpunkt vorhanden sein. Denn nur mit diesen »Hilfsmitteln« ist ein sicheres Schreiben möglich. Versuchen Sie spaßeshalber, mit einer Gummitastatur »blind« zu schreiben. Der Versuch wird nur ein Versuch bleiben. Denn Sie merken bei einer Gummi- oder Folientastatur einfach nicht, ob Sie eine Taste ausreichend stark gedrückt haben. Meist führt das dazu, daß man die Tasten grundsätzlich sehr stark drückt mit den beiden Folgen, daß erstens die Hände schnell ermüden und zweitens die Tastatur schnell kaputtgeht.

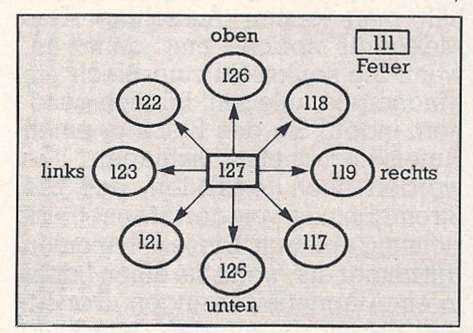


Bild 5. Registerwerte der einzelnen Joystickstellungen

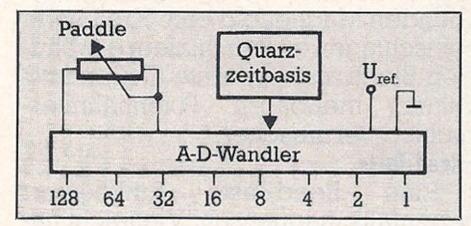


Bild 6. Prinzipschaltbild eines Paddels am A-D-Wandler

Doch nun zur Funktion der Federkontakttaste. Betrachten Sie sich dazu Bild 3, das einen Querschnitt zeigt. Oben sehen Sie die sichtbare Tastenkappe mit einer konkaven Oberfläche. Die Kappe ist über einen Plastikstift mit der eigentlichen Taste verbunden. Häufig lassen sich die Tastenkappen auch abziehen (nicht beim C 64!) und wie ein Druckknopf wieder aufstecken. Der Plastikstift reicht in ein gekapseltes Gehäuse, das eine Verschmutzung der Kontakte durch Staub und Zigarettenrauch verhindert. Innerhalb des Gehäuses ist am Stift eine Kontaktfeder befestigt, welche die beiden Kontaktzungen links und rechts im Gehäuse überbrückt, wenn der Stift tief genug nach unten gedrückt wird. Im unteren Teil befindet sich die Rückstellfeder und die Führung des Stiftes. Zusammengehalten wird die ganze Konstruktion von einer Sicherungsscheibe (Sprengring) am Ende des Plastikstiftes, das aus der Unterseite hinausragt.

Soweit die Mechanik. Aber woher weiß der C 64, welche Taste Sie gedrückt haben? Die Frage kann an dieser Stelle nur prinzipiell beantwortet werden. Zuerst müssen Sie wissen, daß für den C 64 alle Tasten der Tastatur als eine 8x8-Matrix verschaltet sind (Ausgabe 4/84, Seite 115). Jeder Taste kann also ein Spalten- und Zeilenwert innerhalb dieser Matrix zugeordnet werden. Da der C 64 nun mal ein 8-Bit-Computer ist, haben die Spalten und Zeilen die Wertigkeiten 254, 253, 251, 247, 239, 223, 191, 127 (die invertierten Binärzahlen von 1, 2, 4, 8, 16, 32, 64, 128). Wird eine Taste gedrückt, werden die Spalten- und Zeilenzahl der Taste in zwei Registern (Speicherzellen $DC00 und $DC01) des C 64 geschrieben. Werden mehrere Tasten gleichzeitig gedrückt, werden die binären Wertigkeiten einfach addiert. Nun ist es aber nicht möglich, mit ein paar Drähten einfach in ein Register, was weiter nichts ist als eine RAM-Speicherzelle, Werte zu schreiben. Das bedarf schon etwas Aufwand. Im C 64 heißt dieser Aufwand 6526; die CIA 1. Die CIA 1 (Complex Interface Adapter, Vielseitiger Ein-/Ausgabe-Baustein) übernimmt beim C 64 die ständige Abfrage der Tastatur, wertet die geschlossenen Kontakte aus und hält für den Prozessor die Matrixwerte bereit. Wie schon erwähnt, stehen die Werte in den Registern $DC00 und $DC01 (der CIA). Wenn Sie näheres darüber erfahren wollen, empfehlen wir Ihnen den Kurs alle Tasten- und Steuercodes, beginnend ab Ausgabe 4/84. Besonders interessant sind zu diesem Thema die Ausgaben 4/84 und 8/84.

Bestimmt haben Sie auch schon Anzeigen gesehen, in denen Zusatztastaturen angeboten werden. Meist in Form von 10er-Blöcken, die ein schnelles Eintippen von Zahlenkolonnen ermöglichen sollen. Es müssen zwei Versionen unterschieden werden: Software- und Hardwarelösungen.

Die Software-Versionen bestehen aus einer Tastatur, die beim C 64 entweder am Joystick- oder User-Port angeschlossen wird und einem dazugehörigen Maschinenprogramm. Das Maschinenprogramm fragt dann ständig ab, ob eine Zusatztaste gedrückt ist. Wenn ja, teilt es dem Prozessor, wie die serienmäßige Tastatur, den Spalten- und Zeilenwert der Taste mit.

Der Vorteil der Software-Tastaturen liegt in der Einfachheit des Anschlusses, der keinerlei handwerkliche Fertigkeiten verlangt. Stecker in den Joystick-Port stecken, Computer einschalten, Programm laden und einen SYS-Befehl eintippen. Fertig. Der entscheidende Nachteil dieser Versionen liegt an der benötigten Software: Das Laden des Steuerprogramms kostet Zeit. Und das nicht nur beim Einschalten. Denn STOP/RESTORE kann das Steuerprogramm auch abschalten, was die erneute Eingabe des SYS-Befehls zur Initialisierung bedeutet. Außerdem funktioniert die Tastatur überhaupt nicht, wenn Ihr Textprogramm genau an der Speicherstelle des Steuerprogramms steht. Selbst wenn sich das Textprogramm in einem anderen Speicherbereich befindet, kann es immer noch sein, daß der eingegebene Text das Steuerprogramm überschreiben kann. Ein »Absturz« des C 64 wäre dann die Folge.

Die reinen Hardwaretastaturen werden an der C 64-Platine angeschlossen. Der Anschluß selbst ist ein »Klacks«: Lösen der drei Kreuzschlitzschrauben an der C 64-Unterseite, Gehäuse aufklappen, Tastaturstecker links hinten abziehen, Zwischenstecker der Zusatztastatur mit Tastaturstecker aufstecken und das Gehäuse wieder zuschrauben. Allerdings sollten Sie besonders beachten, daß Sie durch das Aufschrauben die halbjährliche Garantie des C 64 verlieren. Dafür stellen



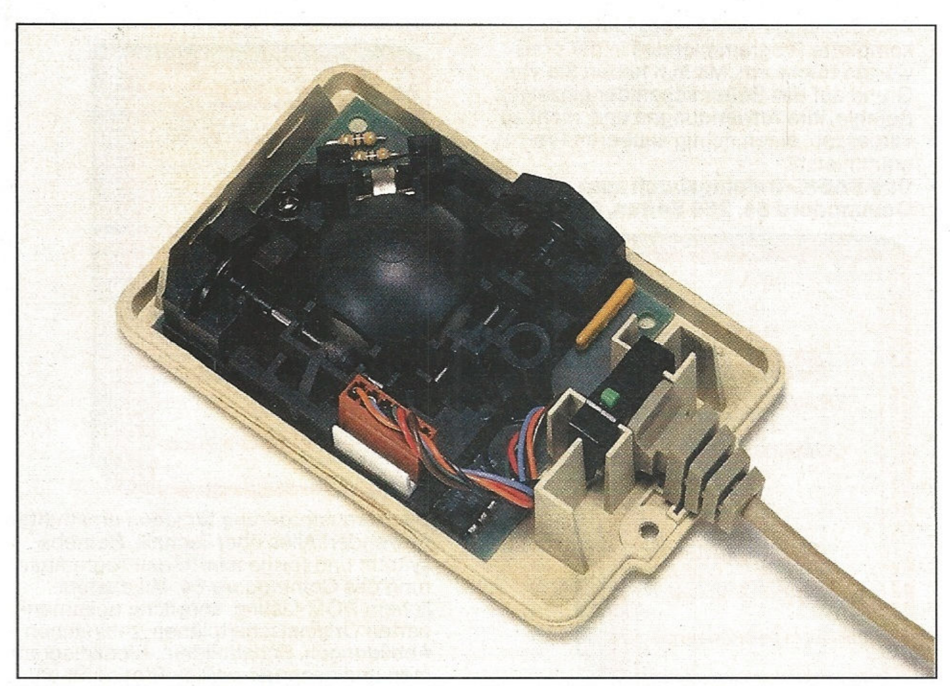


Bild 7. Die Mechanik und die Elektronik einer Maus ist identisch mit der eines Trackballs

die Hardwaretastaturen aber die ideale Lösung dar. Keine Software-Probleme und 100prozentige Verträglichkeit mit allen Programmen. Denn diese Zusatztastaturen überbrücken die gleichen Leitungen wie die Standardtastatur. Es versteht sich fast von selbst, daß dieses Prinzip keine Kompatibilitätsprobleme verursachen kann.

Wer kennt sie nicht, die Schnittstelle Computer — Spielefreak, die Joysticks. Ein Joystick besteht aus vier Schaltern (Bild 4), die auf einer Grundplatte quadratisch angeordnet sind. In der Mitte der Platte ist der Knüppel gelagert. Wird er auf die Seite gedrückt, werden ein oder zwei Schalterkontakte geschlossen. Im Knüppel ist meist noch ein Schalter untergebracht, über den einem Angreifer der letzte Schlag versetzt wird. Die möglichen neun Schalterstellungen (mit Diagonalrichtungen und Feuerknopf) werden von der CIA 1 im C 64 über die fünf Eingän-

Die Schaltknüppel

ge des Joystick-Port abgefragt. Die Eingänge haben normalerweise High-Pegel und werden über die Schalter im Joystick auf Masse gelegt. Wie bei der Tastatur ordnet die CIA 1 den Eingängen verschiedene binäre Wertigkeiten zu. Die Wertigkeiten der Eingänge addiert die CIA und legt den Wert EXOR 255 in einem Register 1 oder 2 der CIA 1 ab:

Port 1: 56321, Port 2: 56320

Die Registerwerte für die möglichen Richtungen entnehmen Sie bitte Bild 5. Alles über Joysticks finden Sie in der nächsten Ausgabe des 64er, in der wir uns ausführlich mit diesem Thema befassen werden.

Joysticks sind die Eingabegeräte der Spielefreaks. C 64-Grafiker müssen sich dieser Knüppel notgedrungen bedienen, da es leider noch keine Maus für den C 64 gibt.

Aus einem Potentiometer wird ein Paddle

Paddles, das sind beim C 64 nichts weiter als einfache Potentiometer (Drehwiderstände) in einem großen Gehäuse. Von diesen etwas exotischen Eingabegeräten lassen sich bis zu vier Stück an den C 64 anschließen. Kennen Sie die alten Telespiele aus den Spielhallen? Aus dieser Zeit stammen die Paddles.

Die Funktion soll an dieser Stelle nur kurz angeschnitten werden. Interessierte können sich im Kurs »C 64 Extern« genauer informieren. Sie finden dort auch eine Bauanleitung dazu. Der Sound-Chip (SID) des C 64 besitzt zwei A-D-Wandler, der die Widerstandswerte der Paddles in Zahlen von 0 bis 255 übersetzt. Die Schaltungsanordnung sieht so aus. Der Drehwiderstand wird an einer Seite mit +5V verbunden, an der anderen Seite mit dem Eingang des A-D-Wandlers. Der Widerstandswert muß so gewählt werden, daß zwischen 0...5V Spannungsabfall geregelt werden kann. Der Widerstandswert bewegt sich dabei zwischen 0 und 500 KOhm. Die digitalen Werte stehen in Register 25 (Adresse 54297) und 26 (54298) des SID. Das Schaltbild eines Paddles finden Sie in Bild 6.

A-D-Wandler

Ein A-D-Wandler dient zur Digitalisierung von analogen Signalen. Zur Digitalisierung lädt man einen entladenen Kondensator mit der Testspannung auf und stellt die Zeit fest, die zur restlichen Aufladung auf die Referenzspannung benötigt wird. Der Zeitwert EXOR 255 gibt das digitale Maß der analogen Spannung. Der ganze Vorgang wird periodisch wiederholt. Da die Zeitermittlung daher exakt sein muß, verwendet man in den meisten Fällen eine Quartz-Steuerung.

Wie kann man vier Paddles an zwei A-D-Wandler anschließen? Nun, ganz einfach. Man schließt abwechselnd immer nur zwei an. Das Umschalten geschieht über die beiden CIA-Bausteine.

Die elektronische Maus

Neben einer Tastatur ist die »Maus« bei Personal Computern ein sehr verbreitetes Eingabegerät. Die meisten dieser »Hardware-Tierchen« funktionieren wie ein Trackball (Ausgabe 8/85). Die wenigsten optoelektronisch, deren Bauweise später behandelt wird. Nun zu den Trackballschwestern, den »Rollmäusen«. Der hauptsächliche Unterschied zum Trackball liegt in der Art, wie eine Maus oder ein Trackball in der Hand gehalten wird. Beim Trackball dreht man mit der Hand eine Billardkugel, die in einem Gehäuse untergebracht ist. Bei der Maus hat man bildlich gesehen das Trackballgehäuse, mit der Kugel nach unten, in der Hand und bewegt es auf der Tischoberfläche. Man hat dadurch ein wesentlich besseres Gefühl für die Bewegung als beim Trackball. Da man das ganze Gehäuse in die Hand nimmt, ist eine Maus auch kleiner als ein Trackball. Die Kugel hat die Größe einer großen Murmel und besteht aus einer Art Hartgummi. Es muß ja ein guter Reibungskontakt zum »Rollfeld« vorhanden sein.

Die Drehbewegung der Kugel überträgt sich in der Maus auf drei Achsen, die ein rechtwinkliges Dreieck bilden (Bild 7). Von den drei Achsen sind nur die beiden Katheten (die Dreieckschenkel, die senkrecht zueinander stehen) interessant. Die schräg stehende Achse (Hypothenuse) dient nur zur Stabilisierung. An jedem Ende einer Kathetenachse ist eine Schlitzscheibe aufgepreßt. Zwei Gabel-Lichtschrankenpaare registrieren die

Drehgeschwindigkeit der Schlitzscheiben. Je schneller sich eine Achse dreht, desto mehr Impulse liefert die dazugehörige Lichtschranke. Aus der Impulszahl der Lichtschranke für horizontale Bewegung und der für vertikale Bewegung läßt sich genau die Bewegungsrichtung und Geschwindigkeit der Maus ermitteln. Beispiel: Bewegt man eine Maus horizontal, dreht sich nur die »horizontale« Achse und umgekehrt. Bei einer 45-Grad-Diagonalbewegung drehen beide Achsen gleich schnell; die beiden Gabellichtschranken liefern die gleiche Impulszahl. Die Drehrichtung der Achsen liefert ein Vergleich der zwei Impulsflanken eines jedes Lichtschrankenpaares. Wie das genau funktioniert, können Sie in Ausgabe 8/85 bei der Beschreibung des Trackballs nachlesen.

Eine Maus wird am Heimcomputer wie ein Joystick angeschlossen. Eine Elektronik verteilt deshalb die Impulse der Lichtschranken auf die entsprechenden vier Eingänge (oben, unten, rechts, links) des Joystick-Ports am C 64. Zeichnen Sie einmal in einem Malprogramm mit einem Joystick und einem Trackball. Beim Zeichnen von schrägen Linien werden Sie schnell den Unterschied zwischen beiden Eingabegeräten feststellen: Mit dem Joystick lassen sich nur 45-Grad-Diagonalen zeichnen, während mit dem Trackball oder einer Maus auch andere Winkel möglich sind. Wenn Sie mit dem Joystick eine 20-Grad-Diagonale zeichnen wollen, müssen Sie den Knüppel etwa nach dem Schema »zwei links, eins nach oben« bewegen. Das ist der Vorteil der Maus oder des Trackballs.

Eine Maus verfügt selbstverständlich über einen Feuerknopf, mit dem zusätzliche Funktionen aufgerufen werden können. Häufig sind sogar zwei Taster vorhanden, die unabhängig voneinander Kontakte schalten.

Maus mit Augen

Die Ermittlung der Bewegungsrichtungen kann auch anders erfolgen, was die optischen Mäuse zeigen. Diese Mäuse werden auf einer speziellen Matte mit Rasterlinien bewegt. Zwei Reflexlichtschranken, die rechtwinklig zueinander an der Unterseite angebracht sind, zählen die »überfahrenen« Rasterlinien und liefern pro Linie einen Impuls.

Normalerweise nimmt man eine Maus zum Zeichnen von Grafiken mit dem Computer oder zum Anwählen von Menüpunkten. Es macht richtig Spaß, mit einer Maus Grafiken mit einem Computer zu zeichnen. Besonders dann, wenn alle Funktionen über Menüsteuerung mit der Maus ausgewählt werden können, da man dann die Maus ständig in der Hand hat. Negativ ist dabei der große Platzbedarf (etwa 30 x 30 cm), den eine Maus benötigt.

Leider gibt es für den C 64 noch keine Maus zu kaufen. Momentan muß man sich noch selbst behelfen und Mäuse von anderen Computern an den C 64 anpassen. Für den C 128 wird es eine Maus von Commodore geben.

Lichtgriffel & Co.

Wie ein Lichtgriffel und ein Grafiktablett funktioniert, ist in Ausgabe 8/85 genau beschrieben. Ein Lichtgriffel oder englisch Lightpen, besteht aus einem Fototransitor. Mit diesem läßt sich der Strahlverlauf eines Fernsehers oder Monitors abtasten. Trifft der Strahl auf den Fototransistor, schaltet dieser kurzzeitig durch und liefert einen Stromimpuls. Da ein Computer weiß, in welcher Zeile und Spalte der Kathodenstrahl sich befindet, kann der Ort des Lichtgriffels am Bildschirm leicht ermittelt werden.

Lichtgriffel

Der Lichtgriffel eignet sich am besten zum Auswählen von Menüpunkten. Als Grafikeingabegerät scheint er weniger zu taugen, da meist die Abfrage der Ortskoordinaten zu langsam und zu ungenau erfolgt. Die Auswertung der Impulse vom Fototransistor ist bei Heimcomputern so träge, daß eine vernünftige Schreibgeschwindigkeit nicht möglich ist. Vor dem Fototransistor befindet sich eine Linse, die die Empfindlichkeit des Transistors auf einzelne Bildpunkte fokussieren soll. Leider haben aber die verschiedenen Bildröhren unterschiedliche Glasstärken, so daß eine exakte Fokussierung nicht zu erreichen ist. Die Folge ist, daß der Lichtgriffel auf mehrere Punkte gleichzeitig anspricht.

Fazit

Die Tastatur wird sicherlich in den nächsten Jahren noch das Eingabe-Medium Nummer eins bleiben. Alternativen zeigen sich mit Grafik-Tabletts, Mäusen und Joysticks. Dabei dürfte sich die Maus über kurz oder lang auch auf dem Heimcomputer-Sektor durchsetzen. Für den C 128 ist sie ja bereits angekündigt. Die Spracheingabe ist in nächster Zukunft für die kleineren Systeme Utopie. (hm)

Chemieprogramme kostenlos

Der Arbeitskreis Computer im Chemieunterricht versucht seit etwa sechs Jahren, Programmierarbeiten im Fach Chemie zu koordinieren, um zu verhindern, daß zuviele Programme zum gleichen Thema geschrieben werden. Außerdem werden diese Programme im Unterricht getestet und eventuell überarbeitet.

Gesucht werden computerbegeisterte Lehrer oder Schüler, die entweder schon Chemieprogramme besitzen oder bearbeiten oder einfach beabsichtigen, Computerprogramme im Bereich Chemie zu erstellen. Außerdem sollten Sie bereit sein, Ihre Programme kostenlos (eventuell gegen eine kleine Unkostenerstattung) in den Programmpool einzubringen. Derzeitiger Bestand: 300 Programme.

Was geschieht mit den Programmen? Sie werden gesichtet und dann in einem Programmkatalog, der beim Arbeitskreis kostenlos angefordert werden kann, aufgenommen. Aus diesem Programmkatalog kann dann jeder (auch Interessierte, die selbst keine Programme beigesteuert haben) Programme beim »Chemischen Institut Dr. Flad« bestellen. Dazu muß man lediglich die Programmwünsche und eine Diskette einschicken (Rückporto bitte nicht vergessen). Auf diese Weise wurden allein im Jahr 1984 rund 25000 Programme **verschenkt.** (aa)

Info: Arbeitskreis Computer im Chemieunterricht, Chemisches Institut Dr. Flad, Breitscheidstr. 127, 7000 Stuttgart 1

Existieren noch weitere derartige Arbeitskreise für verschiedene Bereiche? Wenn ja, dann bitte bei der 64'er-Redaktion melden.

Computerfirmen als »Menschen für Menschen«

Mit Fußball der guten Sache dienen. Das haben sich die Amateure von sechs Computerfirmen vorgenommen. Am Samstag, den 21.9.1985 ab 13 Uhr, findet zu diesem Zweck im Turu-Stadion in Düsseldorf an der Feuerbachstraße auf'm Hennekamp ein Mini-Fußballturnier statt, an dem folgende Mannschafen teilnehmen:

Commodore Büromaschinen, Frankfurt,
Schneider Computer Division, Türkheim,
Atari Corp. Deutschland, Frankfurt,
Epson Deutschland, Düsseldorf,
Victor Technologies, Frankfurt,
Data Becker, Düsseldorf.

Alle Einnahmen fließen der Äthiopien-Hilfe »Menschen für Menschen« (Initiator: Karl-Heinz Böhm) zu. Dabei sollen die Eintrittsgelder von 5 Mark pro Zuschauer noch aufgestockt werden durch den Erlös aus Verlosung und Spendensammlung. Die Zusammensetzung der Mannschaften (Commodore kontra Atari) läßt auf spannende Matches hoffen. Wird der Marktführer auch Turniersieger? (aa)

Das Doppelleben des Joystick-Ports

Am Joystick- oder Control-Port des C 64 können auch dezimale oder hexadezimale Zusatztastaturen angeschlossen werden. Als Beispiel für 10er-Tastaturen haben wir für Sie die Tastatur von Rushware getestet.



Zusatztastaturen von verschiedenen Computern lassen sich über den Joystick-Port an den C 64 anschließen. (Oben rechts die Zehnertastatur von Rushware.)

Der C 64 und VC 20 und die Atari-Computer haben identische Anschlußbuchsen für Joysticks. Aber nicht nur Joysticks können hier angeschlossen werden, sondern auch so manches Zusatzgerät verschiedener Hersteller. Die Palette der für diese Ports vorgesehenen Geräte ist enorm groß. Joystick und Grafiktablett kennt fast jeder, aber auch Trackball, Sketch Pad, Paddles und eben Zusatztastaturen. Da man bei der Steckerform und der Pinbelegung schon fast von einer Norm sprechen kann, müssen diese Geräte nicht unbedingt das Normenzeichen des Computerherstellers tragen.

Die Zehnertastatur von Rushware nutzt diese Tatsache. Sie soll eine Erleichterung für all diejenigen darstellen, die häufig Zahlenkolonnen tippen müssen. Die Standardbelegung der Tasten erlaubt die komfortable Eingabe der Zahlen 0 bis 9, des Dezimalpunktes, der Grundrechenoperatoren »+, —, *, /« und eines Carriage Return (RETURN-Taste). Die Bezeichnung »Standardbelegung« trifft für die Grundeinstellung der Tastatur zu. Sie können den Tasten nämlich auch andere Funktionen zuordnen. Ein bedienungsfreundliches Hilfsprogramm auf Diskette oder Kassette macht diese »Umbelegung« auch dem Laien möglich. Es ist denkbar, daß man sich aus 16 Tasten eine Art Hexadezimal-Tastatur definiert. Den Tasten können aber nicht nur einzelne Zeichen zugeordnet werden, sondern ganze Wörter. Die selbstdefinierten Belegungen können gespeichert und immer wieder geladen werden.

Auf der Haupttastatur wurden drei Tasten neue Funktionen zugeordnet:

@ Bildschirm löschen
← Return L Komma

Wenn auch die Software die Zusatztastatur recht vielseitig macht, ist sie doch auch ein Schwachpunkt. Denn zur Aktivierung muß erst ein Programm geladen und initialisiert werden. Die Initialisierung ist leider auch nach STOP/RESTORE nötig. Ein anderes Problem ergibt sich bei der Verwendung von professionellen Programmen, die eventuell im gleichen Speicherbereich liegen wie die Tastaturdecodierung oder dort Daten speichern. Der Beschreibung läßt sich nicht entnehmen, zwischen welchen Adressen die Treiberroutine liegt.

Plus und Minus

Der genannte Schwachpunkt der Software gilt prinzipiell für alle Zusatztastaturen, die am Joystick-Port oder am User-Port angeschlossen werden und ein Treiberprogramm benötigen. Die Treibersoftware kann nur durch eine Parallelschaltung der Zusatz- zur Haupttastatur vermieden werden. Der Anschluß dazu erfolgt links auf der C 64-Platine am Tastaturstecker. Der Nachteil dieser Lösung soll aber auch nicht verschwiegen werden: Eine selbstdefinierte Belegung der Tasten muß hardwaremäßig erfolgen, falls die Zusatztastatur das überhaupt erlaubt. Denkbar ist dabei die Einstellung über Lötbrücken oder Jumper, was nicht gerade jedermanns Sache ist. Die Tasten können bei dieser Lösung selbstverständlich nur noch mit einem einzelnen Zeichen belegt werden und nicht mehr mit ganzen Wörtern. Hierfür wäre ein spezielles Treiberprogramm nötig oder eine intelligente Tastatur mit eigenem Speicher und Prozessor. Wegen des Preises ist eine intelligente Tastatur aber für einen Heimcomputer indiskutabel.

Zum Preis von 99 Mark ist die 10er-Tastatur von Rushware eine sinnvolle Ergänzung für denjenigen, der häufig Zahlenkolonnen im Dezimalsystem eingeben muß und den C 64 dafür nicht zerlegen will.

(Arnd Wängler/hm)

Info: Rushware Microhandels GmbH, An der Gümpgesbrücke 24, 4044 Kaarst 2

Spielen auf zwei Bildschirmen

Eine einfache Lösung für ein kompliziert scheinendes Problem bietet dieses Programm. Wie realisiert man auf zwei Bildschirmhälften ein voneinander unabhängiges Scrolling in vier Richtungen?

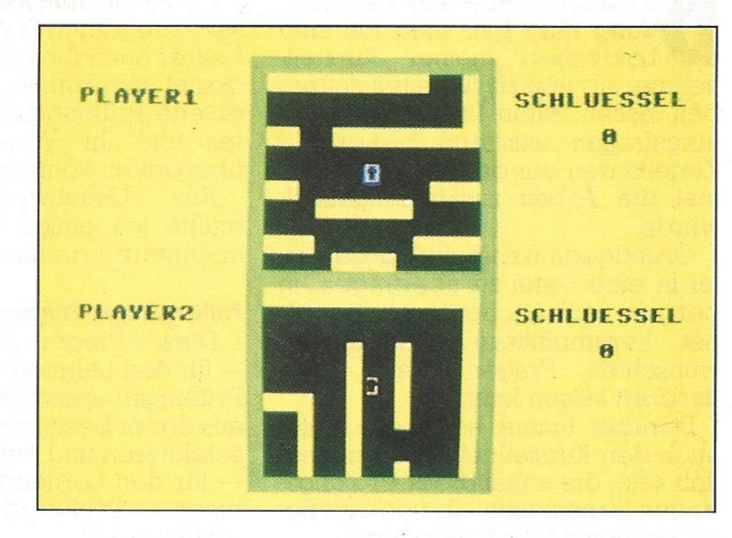
Lebenslauf

Ich wurde am 05.05.69 geboren, bin also 16 Jahre alt, und gehe auf das Gymnasium Moosach in die 10. Klasse.

Meine Liebe zum Computer entdeckte ich vor vier Jahren. Ich sah damals eine Anzeige der Firma Sinclair für den ZX81. Sofort fragte ich meinen Vater, ob ich mir einen kaufen dürfte. Die Antwort war ein klares und deutliches »Nein« (Videospiel-Image, sitzt dauernd vor'm Fernseher etc.). Also heimlich. Eine Woche später besaß ich den ZX81. Nach einem halben Jahr versteckten Programmierens gab ich ihn zurück und gestand meinem Vater meine Eigenmächtigkeit. Aber ich zeigte ihm, daß ich richtig programmierte und kein Spielertyp geworden bin. Daraufhin erlaubte er mir, für damals noch 1400 Mark den neuerschienenen C 64 mit einem Kassettenrecorder zu kaufen. Ein halbes Jahr später folgte ein Diskettenlaufwerk. Seitdem programmiere ich für Firmen und Verlage.



(Andreas von Lepel)



So sieht der zweigeteilte Bildschirm aus, der ein Scrolling in vier Richtungen erlaubt.

Das Programm »Schlüssel« ist ein Spiel für zwei Personen, bei dem derjenige gewinnt, der zuerst drei von fünf Schlüsseln mit seinem Schlüsselloch aufgesammelt hat. Jeder Speicher hat seinen eigenen, unabhängigen Bildschirmausschnitt, der stufenlos in alle vier Richtungen scrollt.

Das Programm liegt ab dez. 26250 und ist 5 KByte lang. Es wird mit SYS26250 gestartet und über zwei Joysticks gesteuert.

Das Besondere an »Schlüssel« ist das voneinander unabhängige Scrollen von zwei Bildschirmhälften in vier Richtungen. Um dies zu erreichen, generiert das Programm vier verschiedene Zeichensätze (einen für jede Richtung). Wenn zum Beispiel eine Bildschirmhälfte nach links gescrollt werden soll, wird in dieser Hälfte auf den für links vorgesehenen Zeichensatz umgeschaltet. Um nun den Bildschirmausschnitt um 2 Bit nach links zu scrollen (um 2 Bit, da es sonst zu langsam geht), wird der gesamte Bildschirmausschnitt durchinkrementiert. Der Zeichensatz »links« ist so geordnet, daß nach jedem Zeichen ein um 2 Bit verschobenes Zeichen folgt. Nachdem auf diese Weise die Bildschirmhälfte dreimal durchinkrementiert wurde (was einer Verschiebung um 6 Bit entspricht), wird der Zeichensatz wieder auf normal zurückgeschaltet und der Bildschirmausschnitt um ein Zeichen nach links gescrollt (Blockscrolling). Somit entsteht der Eindruck eines stufenlosen Scrollings.

Der Vorteil dieser Methode ist die einfache Programmierung und die Schnelligkeit der Programmausführung. Der Nachteil ist, daß für jedes Zeichen, das auf dem Bildschirm gescrollt werden soll, 20 Zeichen pro Zeichensatz benötigt werden (vier Zeichen für jede Richtung und vier Zeichen für die jeweilige Position). Somit kann man also nur insgesamt zwölf verschiedene Zeichen darstellen, was die etwas eintönige Grafik erklärt.

Zum Spiel selbst. »Schlüssel« ist vollkommen per Interrupt gesteuert. Der Hauptprogrammteil (Bewegung, Scrolling, Kollision etc.) liegt ab $668A (26250) bis $7000. Die Grafik hat ein Ausmaß von 40 x 40 Zeichen und liegt ab $7000 bis $7640. Sie wird vor jedem neuen Spiel nach $8000 verschoben, da die Grafik während des Spielens verändert wird. Von $7640 bis $7902 liegt das Unterprogramm, das die vier Zeichensätze generiert.

Das Zeichensatzscrolling kann relativ leicht in eigene Programme integriert werden. Wie man an dem Programm »Schlüssel« sieht, muß es nicht immer ein langes Programm sein, das zum Listing des Monats gewählt wird. Auf die Idee kommt es an. (Andreas von Lepel/rg)

Prüfungsfragen

Prüfungen sind für viele ein Greuel, sowohl für die Lehrer als auch für die geplagten Schüler. Beiden kann mit diesem Programm geholfen werden.

Da ich in der Aus- und Weiterbildung tätig bin, muß ich zum Leidwesen meiner Kursteilnehmer ab und zu Arbeiten schreiben lassen. Mein Bestand an Prüfungsfragen existierte bisher auf Karteikarten, aus denen dann jedesmal die Arbeit zusammengestellt wurde.

Grundgedanke ist also, diese Kartei in eine Datei zu übertragen, in der ich, nach Sachgebieten geordnet, herumblättern und die gewünschten Fragen gleich ausdrucken lassen kann.

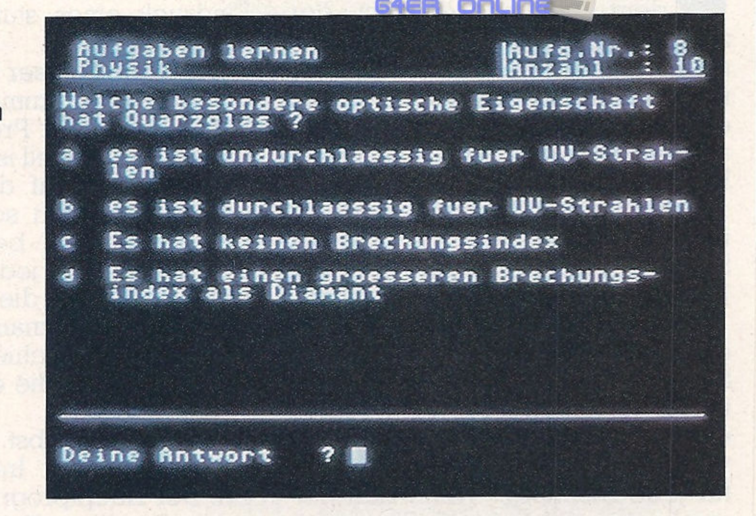
Darüber hinaus soll diese Datei auch den Kursteilnehmern zugänglich sein, die anhand der Prüfungsfragen lernen, nicht aber in den Bearbeitungsmodus hineindürfen.

Ich könnte mir vorstellen, daß dieses Programm neben manchem Lehrer auch für viele Schüler interessant sein könnte, die sich so ihre eigene Prüfungsfragendatei erstellen und ihr Wissen regelmäßig überprüfen können.

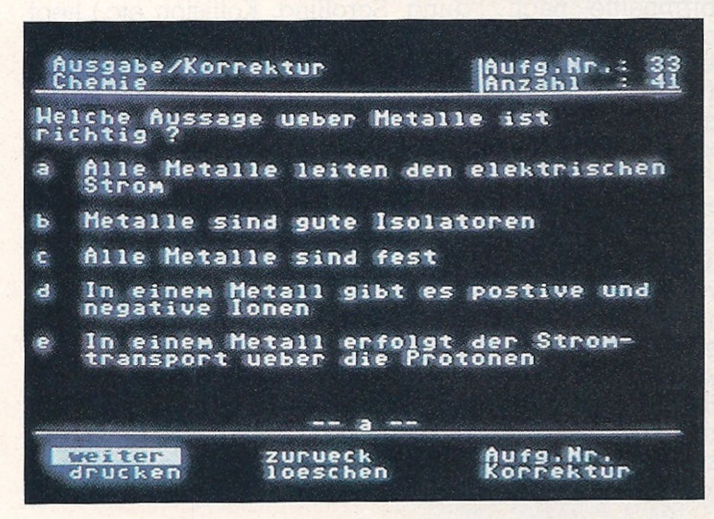
Aus Geschwindigkeitsgründen mußte ich einige Maschinenspracheunterprogramme einbinden.

Prüfungsfragen/Programmbeschreibung

Dieses Programm ist geeignet...
— für den Lehrenden, der sich eine Prüfungsfragendatei erstellen will, aus der er bestimmte Fragen später selektieren und ausdrucken kann.
— für den Lernenden, der anhand einer Prüfungsfragensammlung »pauken« will.



Der Lernmodus. Hier muß der Lernende Fragen beantworten, die vom Computer gestellt werden.



Der Bearbeitungsmodus. Die Fragen werden mit den Antworten angezeigt und können editiert werden.

Lebenslauf



(Hartmut Gölker)

Am 18.3.1944 wurde ich in Dessau geboren. Nach Schule, Lehre, Bundeswehr, Ingenieurschule (Fachrichtung Chemie) und verschiedenen beruflichen Stationen arbeite ich seit 1971 in der Weiterbildungsabteilung eines Chemieunternehmens.

Mit der »Computerei« beschäftige ich mich hobbymäßig seit zirka drei Jahren. Mein Ersterwerb war ein TI 99, dessen Produktion (leider) eingestellt wurde, so daß ich alternativ zum C 64 gekommen bin. Die Möglichkeiten, die in diesem Gerät versteckt sind, werden mich sicher noch für ein paar Jahre in meiner Freizeit voll auslasten.

Eine grundsätzliche Schwierigkeit liegt im Vergleich der Antworten. Die jeweilige Antwort muß »eindeutig« sein. Gut geeignet ist das Multiple-choise-System, das heißt von vorgegebenen Antworten ist eine richtig oder falsch. Möglich ist aber auch die Eingabe von Jahreszahlen, Namen etc., bei denen kaum unterschiedliche Möglichkeiten in der Eingabe vorkommen können.

Nach dem Laden des Programms und Starten mit RUN werden zuerst einige Maschinenspracheprogramme nachgeladen. Sie werden bewußt nicht zu einem Programm zusammengefaßt, da sie in vielen Programmen als Programmbausteine Verwendung finden können.

Das Programm Prüfungsfragen ist menügesteuert, wobei die einzelnen Menüpunkte mit den Cursorsteuertasten angewählt und mit der F1- oder RETURN-Taste übernommen werden können.

(Hartmut Gölker/gk)



Prüfungsfragen

Das Programm erleichtert nicht nur dem Lehrer die Zusammenstellung von Aufgaben, sondern auch dem Kursteilnehmer deren Beantwortung.

Das Programm (Listing 1) ist menügesteuert und erlaubt nicht nur das Auswählen und Beantworten von Fragen, sondern auch die Erstellung eines eigenen Fragenkatalogs nach dem Multiple Choise System. Sie kennen sicher alle entsprechenden Fragebögen: Es wird eine Frage gestellt, und Sie müssen aus mehreren vorgegebenen Antworten die richtige(n) heraussuchen.

Alle Menüpunkte können mit den Cursor-Tasten angewählt und mit der F1- oder RETURN-Taste übernommen werden.

Merken muß man sich noch die F7-Taste, mit ihr gelangt man in den bisherigen Programmabschnitt zurück. Zuerst erscheint das Menü:

— Druck.einst. Einstellung auf die Drucker Epson FX-80 oder Commodore MPS 801
— Beschreibung ist eine Kurzbeschreibung des Programms,
— Start die Aufforderung ... »Aufgabendiskette einlegen« erscheint.

Nach dem Einlegen der Aufgabendiskette, beziehungsweise einer neu formatierten Diskette, kann die eigentliche Arbeit losgehen.

Es folgen die Menüpunkte:
— Lernen ist für den Lernenden zum Pauken von Prüfungsfragen
— Bearbeiten dient unter anderem zur Eingabe von Prüfungsfragen und zur Druckerausgabe

Lernen

Sind die Prüfungsfragen in unterschiedliche Sachgebiete unterteilt, so werden die Sachgebiete in einem Menü gezeigt, aus dem die Auswahl zu treffen ist.

Nach dieser Wahl erscheint die erste Frage. Die Antwort ist einzugeben und mit F1 oder RETURN zu bestätigen. Bei falscher Antwort antwortet der Computer mit einem blinkenden *falsch* und der Anwender hat zwei weitere Eingabemöglichkeiten.

Eine korrekte Antwort wird mit *richtig* quittiert und die nächste Frage kann von der Diskette eingelesen werden. Beim ersten Mal richtig beantwortete Fragen werden aussortiert und erscheinen nicht mehr.

Die Reihenfolge der Fragen ist zufällig. Um den Programmabschnitt zu beenden, muß ich entweder alle Fragen des Sachgebiets richtig beantworten oder mit F7 abbrechen. Danach kommt die Auswertung, die angibt, wieviel Fragen bearbeitet und wieviel davon beim ersten Mal richtig beantwortet wurden.

Bearbeiten

Dieser Programmabschnitt ist durch eine Codenummer geschützt. Als Codenummer wurde 001 festgelegt, das läßt sich aber beliebig ändern. Legt man Wert auf diesen Schutz, muß noch ein LIST-Schutz und/oder ein AUTOSTART eingebaut werden.

Es folgt ein Auswahlmenü der vorhandenen Sachgebiete und den Optionen »Sachgebiet anlegen« und »Sachgebiet löschen«

Sachgebiete
Physik
Chemie
Mathematik
— Sachgebiet anlegen
— Sachgebiet löschen
— Sachgebiet anlegen
— Es können bis zu zehn Sachgebiete angelegt werden, es ist jedoch nicht sinnvoll auf einer Diskette zuviel Sachgebiete unterbringen zu wollen, da die Speicherkapazität begrenzt ist.
— Sachgebiete löschen

►

Checksummer 64 V3 und MSE

Bei beiden Programmen handelt es sich um Eingabeerleichterungen unserer Listings. Ferner werden alle Tippfehler bemerkt und angezeigt.

Mit dem Checksummer 64 V3 läßt sich überprüfen, ob eine Zeile korrekt eingegeben wurde. Dazu muß zuerst das Programm Checksummer 64 V3 geladen und mit RUN gestartet werden. Ist das geschehen, erscheint, sobald Sie eine Zeile eingeben und mit RETURN abschlossen haben, links oben auf dem Bildschirm eine geklammerte Zahl in reverser Darstellung. Bei dieser Zahl handelt es sich um eine Prüfsumme. Sie muß mit der ebenfalls geklammerten Zahl am Ende jeder Basic-Zeile übereinstimmen. Tut sie das nicht, haben Sie einen Tippfehler gemacht. Die Prüfsummen im Listing dürfen nicht mit abgetippt werden. Basic-Listings enthalten keinerlei Steuer- und Grafikzeichen, dafür aber unter- beziehungsweise überstrichene Zeichen und Wörter in geschweiften Klammern. Dabei bedeuten:
unterstrichenes Zeichen = SHIFT-Taste + Zeichen
überstrichenes Zeichen = Commodore-Taste + Zeichen
die Wörter in geschweiften Klammern:
UP, DOWN, RIGHT, LEFT = Cursorsteuertasten

Ferner werden alle Farben ausgeschrieben. Die Farbe, deren Abkürzung Sie auf der Tastatur des C 64 wiederfinden, erreichen Sie über die Control-Taste (CTRL) beziehungsweise Commodore-Taste plus der Farbtaste

Zahlen in geschweiften Klammern geben an, wie oft die Tastenkombination unmittelbar hinter der Zahl zu drücken ist, zum Beispiel 2SPACE = 2 * Leertaste.

Das Listing zu diesem Checksummer V3 finden Sie zum erstenmal in der Ausgabe 8/85, Seite 54, und dem Drucker/Grafik-Sonderheft. Alle anderen Checksummer-Listings (V2) sind damit hinfällig. Die ausführliche Anleitung aus diesen Ausgaben kann jedoch weiterhin benutzt werden.

Der **MSE** unterstützt die Eingabe von Listings, die in Maschinensprache geschrieben wurden. Bei ihm handelt es sich um ein Maschinensprache-Editor, der Fehleingaben ausschließt. Eine abgetippte Zeile wird nur dann angenommen, wenn sie richtig eingegeben wurde.

Das Listing zum MSE mit ausführlicher Anleitung finden Sie in den Ausgaben 1/85 bis 6/85. Auch dieses Programm ist auf jeder Leserservice-Diskette gespeichert.

Außerdem schicken wir Ihnen gegen Einsendung eines an Sie selbst adressierten Briefumschlags (Größe DIN C5 = 0,80 Mark, DIN C4 = 1,10 Mark Porto) gerne das Listing zum MSE und Checksummer 64 V3 zu.

— Das Sachgebiet und alle zugehörigen Prüfungsfragen werden auf der Diskette gelöscht.

Ist ein Sachgebiet gewählt, muß entschieden werden, ob neue Fragen eingegeben oder vorhandene bearbeitet werden sollen:

— Eingabe — Auf dem bis auf Kopf- und Fußzeilen leeren Bildschirm wird eine Frage eingegeben. Der Cursor kann sich innerhalb des vorgegebenen Raumes frei nach oben und unten bewegen.

Mit F1 wird der Text übernommen, und nachdem nach Aufforderung die Antwort eingetippt worden ist, wird die Frage gespeichert.

In der rechten oberen Ecke des Bildschirms erscheint jeweils die Anzahl der bisher gespeicherten Fragen und die Nummer der in Arbeit befindlichen Frage.

— Ausgabe — Die erste Frage des angewählten Sachgebietes erscheint automatisch. Danach folgt in den Fußzeilen das Menü:

weiter	zurück	Aufg.Nr.
drucken	löschen	Korrektur

Die Maschinensprach-Unterprogramme

1. Print

Die Routine erlaubt die positionierte Bildschirmausgabe (Listing 2). Syntax: sys pr, Spalte, Zeile, Druckliste

2. IN21

Mit dieser erweiterten Input-Routine kann eine Variable an einer bestimmten Stelle des Bildschirms eingelesen werden (Listing 3).

Syntax: sys in, Spalte, Zeile, Länge, Art, Variable

— Länge = Maximallänge
— Art = numerische (0)/alphanumerische (1)
— Variable = Variable, in die der String eingelesen wird.

3. Bildschirm2

Die Routine erlaubt eine freie Texteingabe innerhalb eines bestimmten Bildschirmbereichs von Spalte1/Zeile1 bis Spalte 2/Zeile2 (Listing 4). Syntax: sys bi, sp1, ze1, sp2, ze2

4. READ10. Die Routine liest einen Text vom Bildschirm, wobei alle überflüssigen Leerzeichen entfernt werden und überträgt den Text wahlweise auf die Diskette oder den Drucker (Listing 5). Ein zweiter Einspringpunkt liest umgekehrt den Text von der Diskette und gibt ihn wieder in der ursprünglichen Form auf dem Bildschirm aus.

5. Drucker. Mit dieser Routine läßt sich vom Programm her überprüfen, ob der Drucker eingeschaltet ist. Sie stammt aus dem 64'er-Magazin (Listing 6).

(Hartmut Gölker/gk)

```
10 REM PROGRAMMIERTE FRAGEN                          <199>
20 POKE 53280,12:POKE 53281,0                        <176>
100 PRINT CHR$(147)                                  <129>
101 IF A=0 THEN A=1:LOAD"PRINT OBJ",8,1              <053>
102 IF A=1 THEN A=2:LOAD"IN21 OBJ",8,1               <047>
103 IF A=2 THEN A=3:LOAD"BILDSCHIRM OBJ",8
 ,1                                                  <059>
104 IF A=3 THEN A=4:LOAD"READ10 OBJ",8,1             <244>
105 IF A=4 THEN A=5:LOAD"DRUCKER OBJ",8,1            <151>
110 PR=12*16↑3   : DR=PR+6*16↑2                      <209>
111 IN=PR+1*16↑2 : BI=PR+3*16↑2                      <177>
112 WR=PR+5*16↑2 : RE=WR+13*16+10                    <197>
113 DIM L(99,2),M$(15),SP(15),ZE(15)                 <215>
114 PRINT CHR$(14):POKE 657,128                      <222>
118 AU=0:KO$="PRUEFUNGSFRAGEN":GOSUB 11000
 :REM KOPF                                           <243>
119 SYS PR,01,01,"H. GOELKER / 28.03.84{LI
 G.BLUE}"                                            <083>
125 M$(1)= " DRUCK.EINST. ":SP(1)=0:ZE(1)=
 24                                                  <112>
130 M$(2)= " BESCHREIBUNG ":SP(2)=16:ZE(2)
 =24                                                 <222>
135 M$(3)= " START ":SP(3)=32:ZE(3)=24:NN=
 3:J=2:GOSUB 14010                                   <168>
140 IF X$=CHR$(136) THEN 118                         <217>
145 IF J=3 THEN 156                                  <196>
150 ON J GOSUB 15000,16000                           <231>
151 GOTO 118                                         <096>
156 KO$="PRUEFUNGSFRAGEN":GOSUB 11000                <206>
158 GOSUB 13000                                      <186>
159 IF X$=CHR$(136) THEN 118                         <236>
160 GOSUB 10700:KO$="PRUEFUNGSFRAGEN":GOSU
 B 11000                                             <057>
162 M$(1)=" LERNEN ":SP(1)=7:ZE(1)=24                <042>
164 M$(2)=" BEARBEITEN ":SP(2)=22:ZE(2)=24
 :NN=2                                               <008>
166 GOSUB 14000                                      <210>
168 IF X$=CHR$(136) THEN 118                         <245>
170 ON J GOSUB 200,800                               <097>
180 GOTO 160                                         <204>
190 :                                                <166>
200 REM AUFGABEN LERNEN ***************              <246>
210 KO$="AUFGABEN LERNEN":GOSUB 11000                <165>
230 GOSUB 12500                                      <146>
240 IF X$=CHR$(136)THEN RETURN                       <128>
245 IF F=0 THEN GOSUB 10000                          <010>
250 IF N>0 THEN 300                                  <205>
255 GOSUB 11400                                      <123>
260 SYS PR,07,24,"KEINE AUFGABEN GESPEICHE
 RT"CHR$(145)                                        <004>
270 GET X$:IF X$=""THEN 270                          <182>
280 SA$="":RETURN                                    <249>
300 FOR I=1 TO N:L(I,1)=I:L(I,2)=0:NEXT              <220>
310 MAX=N                                            <229>
320 T=INT(RND(1)*MAX)+1:R=L(T,1)                     <225>
330 IF L(T,2)=0 THEN FR=FR+1                         <127>
340 GOSUB 17000:GOSUB 10600:REM KOPF/FRAGE           <169>
350 SYS PR,00,24,"DEINE ANTWORT{3SPACE}?";           <073>
360 SYS IN,18,24,21,1,ER$                            <204>
365 IF PEEK(828)=136 THEN 700                        <059>
370 IF ER$<>AW$ THEN 500                             <090>
380 IF L(T,2)=0 THEN RI=RI+1                         <078>
390 L(T,1)=L(MAX,1):L(T,2)=L(MAX,2)                  <115>
400 MAX=MAX-1                                        <018>
410 GOSUB 11400:REM FUSSZEILEN LOESCHEN              <116>
420 SYS PR,12,23," ** RICHTIG ** "                   <087>
440 GET X$:IF X$<>CHR$(13)AND X$<>CHR$(136
 )THEN 440                                           <198>
450 IF X$=CHR$(136) OR MAX=0 THEN 700                <171>
460 GOTO 320                                         <182>
470 :                                                <192>
500 REM FALSCHE ANTWORT                              <064>
510 L(T,2)=1                                         <112>
520 GOSUB 12000 :REM FALSCHE ANTWORT                 <116>
530 GOSUB 11400                                      <146>
540 SYS PR,00,24,"DEINE ANTWORT{3SPACE}?";           <009>
550 SYS IN,18,24,21,1,ER$                            <140>
560 IF ER$=AW$ THEN FA=0:GOTO 410                    <185>
570 GOSUB 12000:FA=FA+1:IF FA<2 THEN 530             <131>
580 GOSUB 11400:GOSUB 13200                          <171>
581 SYS PR,(40-L)/2,23," * RICHTIG IST :"A
 W$" * "                                             <031>
600 FA=0: GOTO 440                                   <094>
610 :                                                <078>
700 GOSUB 11000:REM KOPF                             <137>
710 SYS PR,2,12,"ANZAHL DER FRAGEN :";FR             <237>
720 SYS PR,2,14,"RICHTIGE ANTWORTEN:";RI             <044>
730 FA=0:RI=0:FR=0:SA$=""                            <151>
740 GET X$:IF X$="" THEN 740                         <110>
750 RETURN                                           <044>
760 :                                                <228>
800 REM AUFGABEN BEARBEITEN ***********              <024>
810 IF S$="001{2SPACE}"THEN 890                      <061>
820 KO$="AUFGABEN BEARBEITEN":GOSUB 11000            <050>
830 SYS PR,01,05,"BITTE CODE-NR. EINGEBEN:
 "                                                   <111>
840 SYS IN,01,07,5,1,S$                              <189>
850 IF S$="001{2SPACE}"THEN 890                      <101>
860 SYS PR,01,24,"CODE-NR. UNBEKANNT / WEI
 TER MIT RETURN"CHR$(145)                            <222>
870 GET X$:IF X$<>CHR$(13) THEN 870                  <086>
880 GOTO 800                                         <102>
890 GOSUB 11000:REM KOPF                             <073>
900 GOSUB 12200:IF A=136 THEN SA$="": RETU
 RN                                                  <207>
980 :                                                <194>
1000 REM AUFGABEN BEARBEITEN TEIL 2 ---              <096>
```

Listing 1. Das Programm Prüfungsfragen. Beachten Sie die Hinweise zum Abtippen auf Seite 54

```
1010 KO$="AUFGABEN BEARBEITEN":GOSUB 11000
     :REM KOPF                                 <139>
1020 M$(1)=" EINGABE ":SP(1)=6:ZE(1)=24        <135>
1030 M$(2)=" AUSGABE ":SP(2)=24:ZE(2)=24       <180>
1090 NN=2:GOSUB 14000                          <151>
1095 IF X$=CHR$(136) THEN SA$="":RETURN        <017>
1100 ON J GOSUB 1140,3000,1090:GOTO 1000       <022>
1110 :                                         <070>
1140 REM EINGABE -----------------------       <235>
1145 KO$="AUFGABEN EINGEBEN":AW$=""            <254>
1146 IF N+1<99 THEN 1150                       <168>
1147 GOSUB 11400:SYS PR,5,23,"{RVSON}KEINE
     EINGABE MEHR MOEGLICH{RVOFF}":GOSUB
     11100:GOTO 1280                           <049>
1150 R=N+1:GOSUB 11000:REM KOPF                <244>
1160 SYS PR,10,24,"F1=TEXT UEBERNEHMEN"CHR
     $(145)                                    <024>
1170 SYS BI,0,3,39,20                          <149>
1180 A=PEEK(828)                               <032>
1195 IF A=136 THEN 1280                        <184>
1200 GOSUB 11400:REM ZEILE 24 LOESCHEN         <121>
1205 POKE 781,21:SYS 59903                     <200>
1210 SYS PR,0,24,"RICHTIGE ANTWORT: "AW$;C
     HR$(145)                                  <101>
1220 SYS IN,18,24,21,1,AW$                     <058>
1225 A=PEEK(828)                               <077>
1226 IF A=136 THEN 1280                        <215>
1230 A=PEEK(828):IF A=17 OR A=145 OR AW$="
     {21SPACE}"THEN 1220                       <168>
1240 IF KO=0 THEN N=N+1:R=N                    <103>
1260 GOSUB 10500:REM AUFGABEN WRITE            <132>
1270 GOSUB 10200:REM STATUS WRITE              <036>
1280 R=0:RETURN                                <213>
1290 :                                         <252>
3000 REM AUSGABE                               <018>
3010 R=0: KO$="AUSGABE/KORREKTUR":IF N<1 T
     HEN 3700                                  <063>
3015 GOTO 3200                                 <209>
3020 GOSUB 11000                               <222>
3100 GOSUB 10600                               <224>
3102 GOSUB 13200:SYS PR,(52-L)/2,21," --
     AW$" -- "                                 <040>
3105 GOSUB 11400                               <181>
3110 M$(1)=" WEITER ":SP(1)=1:ZE(1)=23         <242>
3111 M$(2)=" ZURUECK{2SPACE}":SP(2)=13:ZE(
     2)=23                                     <171>
3112 M$(3)=" AUFG.NR.{2SPACE}":SP(3)=27:ZE
     (3)=23                                    <098>
3120 M$(4)=" DRUCKEN ":SP(4)=1:ZE(4)=24        <136>
3121 M$(5)=" LOESCHEN ":SP(5)=13:ZE(5)=24      <116>
3122 M$(6)=" KORREKTUR ":SP(6)=27:ZE(6)=24     <026>
3130 NN=6:GOSUB 14000                          <031>
3135 IF X$=CHR$(136) THEN 3800                 <210>
3140 ON J GOTO 3200,3300,3400,5000,4000,36
     00                                        <054>
3150 :                                         <078>
3200 REM WEITER ------------------------       <221>
3210 R=R+1:IF R<=N THEN 3020                   <067>
3220 GOSUB 11400                               <040>
3230 SYS PR,8,24,"ENDE DES AUFGABENSATZES"
     CHR$(145)                                 <080>
3240 GET X$:IF X$=""THEN 3240                  <119>
3250 GOTO 3800                                 <030>
3260 :                                         <188>
3300 REM ZURUECK -----------------------       <232>
3310 R=R-1:IF R>=1 THEN 3020                   <238>
3320 GOTO 3220                                 <068>
3330 :                                         <004>
3400 REM NUMMER EINGEBEN --------------        <073>
3410 R=0:GOSUB 11000                           <097>
3420 SYS PR,1,24,"AUFGABEN-NUMMER ?"CHR$(1
     45)                                       <220>
3430 SYS IN,19,24,3,0,B$:B=INT(VAL(B$))        <152>
3435 IF PEEK(828)=136 THEN 3020                <178>
3440 IF B>N OR B<1  THEN SYS PR,1,23,"UNGU
     ELTIGE NUMMER":GOTO 3420                  <038>
3450 R=B:GOTO 3020                             <044>
3460 :                                         <134>
3600 REM KORREKTUR ---------------------       <098>
3610 GOSUB 11400:REM FUSSZEILEN LOESCHEN       <014>
3615 SYS PR,09,23,"GIB DIE KORREKTUR EIN !
     "                                         <044>
3620 KO=R:GOSUB 1160:R=KO:KO=0                 <117>
3630 IF A=136 THEN RETURN                      <126>
3640 GOTO 3105                                 <121>
3650 :                                         <070>
3700 GOSUB 11000:SYS PR,8,24,"KEINE AUFGAB
     E GESPEICHERT "CHR$(145)                  <051>
3710 GET X$:IF X$=""THEN 3710                  <016>
3800 R=0:RETURN                                <191>
3810 :                                         <230>
4000 REM LOESCHEN ---------------------        <051>
4100 GOSUB 11400                               <160>
4110 SYS PR,4,24,"AUFGABE WIRKLICH LOESCHE
     N (J/N) ?"CHR$(145)                       <114>
4120 GET X$:IF X$<>"J" AND X$<>"N" THEN 41
     20                                        <090>
4130 IF X$="N" THEN 3105                       <232>
4135 GOSUB 11400:SYS PR,5,24,"... AUFGABE
     WIRD GELOESCHT ..."CHR$(145)              <215>
4140 IF N=1 OR N=R THEN 4570                   <106>
4500 OPEN 15,8,15                              <084>
4510 PRINT#15,"S:"STR$(R)+SA$                  <172>
4520 PRINT#15,"R:"STR$(R)+SA$"="STR$(N)+SA
     $                                         <169>
4530 CLOSE 15                                  <050>
4570 N=N-1                                     <111>
4580 GOSUB 10200:REM STATUSDATEI               <235>
4590 GOTO 3800                                 <100>
4600 :                                         <002>
5000 REM DRUCKEN **********************        <172>
5001 GOSUB 11400                               <043>
5002 POKE 2,4:SYS DR                           <172>
5003 IF PEEK(2)=0 THEN AU=AU+1:GOTO 5008       <022>
5004 SYS PR,07,24,"DRUCKER NICHT EINGESCHA
     LTET"CHR$(145)                            <186>
5005 GET X$:IF X$=""THEN 5005                  <045>
5006 GOTO 3105                                 <217>
5007 :                                         <157>
5008 ON D GOTO 5009,5100                       <119>
5009 REM EPSON DRUCKER ----------------        <199>
5010 OPEN 2,4,0:CMD 2                          <111>
5020 PRINT CHR$(27);"3";CHR$(22);:PRINT CH
     R$(15);                                   <160>
5026 PRINT CHR$(27);"L";CHR$(5);               <068>
5027 PRINT   "AUFG.";AU                        <247>
5028 PRINT CHR$(27);"L";CHR$(15);              <024>
5030 SYS WR:CLOSE 2                            <173>
5070 GOTO 3105                                 <025>
5080 :                                         <230>
5100 REM VC 1526 DRUCKER ---------------       <161>
5102 OPEN 2,4,7:PRINT#2,"AUFG.";AU             <057>
5103 PRINT#2,LEFT$("YYYYYYYYYYYYYYY",5+LEN(
     STR$(AU)))                                <012>
5105 OPEN 6,4,6:PRINT#6,CHR$(16);              <223>
5130 SYS WR                                    <139>
5140 PRINT#2:CLOSE 2:CLOSE 6: GOTO 3105        <206>
10000 REM DATEIEN *********************        <150>
10010 :                                        <080>
10020 REM STATUSDATEI READ ------------        <004>
10100 N=0:R=0:OPEN 15,8,15                     <009>
10110 OPEN 2,8,2,SA$+"/S,S,R"                  <088>
10120 INPUT#15,F:IF F<>0 THEN 10140            <029>
10130 INPUT#2,N:REM ANZAHL AUFGABEN            <194>
10140 CLOSE 2:CLOSE 15:RETURN                  <241>
10195 :                                        <009>
10200 REM STATUSDATEI ANLEGEN ---------        <066>
10230 OPEN 2,8,2,"@:"+SA$+"/S,S,W"             <038>
10240 PRINT#2,N                                <206>
10250 CLOSE 2:RETURN                           <225>
10260 :                                        <076>
10300 REM SACHGEBIETE WRITE -----------        <122>
10310 IF II>0 THEN 10350                       <192>
10320 OPEN 15,8,15,"S:SACHGEBIETE"             <122>
10325 SA$(1)=""                                <161>
10330 CLOSE 15:RETURN                          <242>
10340 :                                        <156>
10350 OPEN 2,8,2,"@:SACHGEBIETE,S,W"           <215>
10360 FOR I=1 TO II:PRINT#2,SA$(I):NEXT        <203>
10370 CLOSE 2:CLOSE 15:RETURN                  <217>
10500 REM AUFGABEN WRITE --------------        <007>
10504 GOSUB 11400:REM ZEILE 24 LOESCHEN        <027>
10505 SYS PR,08,24,"... BITTE WARTEN ..."C
      HR$(145)                                 <032>
10510 SYS PR,00,3,"";                          <006>
10520 OPEN 3,3                                 <169>
10530 OPEN 2,8,2,"@:"+STR$(R)+SA$+",S,W"       <007>
10540 SYS WR                                   <215>
10565 PRINT#2,AW$                              <091>
10570 CLOSE 2:CLOSE 3                          <050>
```

```
10580 RETURN                                   <224>
10590 :                                        <152>
10600 REM AUFGABEN READ ---------------        <097>
10610 OPEN 2,8,2,STR$(R)+SA$+",S,R"            <195>
10620 SYS RE                                   <046>
10630 INPUT#2,AW$                              <076>
10660 CLOSE 2:RETURN                           <125>
10700 REM SACHGEBIETE PRUEFEN/EINLESEN         <251>
10705 II=0                                     <195>
10710 OPEN 15,8,15                             <198>
10720 OPEN 2,8,2,"SACHGEBIETE,S,R"             <139>
10730 INPUT#15,F                               <064>
10740 IF F<>0 THEN 10780                       <039>
10750 II=II+1                                  <235>
10760 INPUT#2,SA$(II)                          <181>
10770 IF ST<>64 THEN 10750                     <093>
10780 CLOSE 2:CLOSE 15:RETURN                  <119>
10790 :                                        <098>
11000 REM KOPF -------------------------       <103>
11010 PRINT CHR$(147);:PRINT TAB(1)KO$         <133>
11015 IF SA$="" THEN 11020                     <141>
11016 SYS PR,1,1,SA$                           <136>
11017 SYS PR,28,0,"AUFG.NR.:"R                 <244>
11018 SYS PR,28,1,"ANZAHL{2SPACE}:"N           <213>
11020 SYS PR,00,02,"TTTTTTTTTTTTTTTTTTTTTT
      TTTTTTTTTTTTTTTTTT"CHR$(145)             <238>
11021 SYS PR,27,0,"N"                          <107>
11022 SYS PR,27,1,"N"                          <110>
11030 SYS PR,00,22,"TTTTTTTTTTTTTTTTTTTTTT
      TTTTTTTTTTTTTTTTTT"CHR$(145)             <252>
11040 SYS PR,00,03,"";                         <005>
11050 RETURN                                   <186>
11060 :                                        <114>
11100 REM WEITER MIT RETURN -----------        <246>
11110 SYS PR,10,24,"WEITER MIT RETURN"CHR$
      (145)                                    <027>
11120 GET X$:IF X$<>CHR$(13) THEN 11120        <187>
11130 A=136:RETURN                             <027>
11140 :                                        <194>
11400 REM ZEILE 24 LOESCHEN -----------        <229>
11410 POKE 781,23:SYS 59903                    <247>
11420 POKE 781,24:SYS 59903:RETURN             <089>
11430 :                                        <230>
11580 :                                        <126>
12000 REM FALSCHE ANTWORT -------------        <112>
12010 FOR J=1 TO 5                             <178>
12020 SYS PR,25,24,"** FALSCH **";             <159>
12030 FOR JJ=1 TO 60:NEXT                      <220>
12040 SYS PR,25,24,"{RVSON}** FALSCH **{RV
      OFF}";                                   <198>
12050 FOR JJ=1 TO 60:NEXT:NEXT                 <229>
12060 RETURN                                   <180>
12070 :                                        <108>
12200 REM SACHGEBIET WAEHLEN/ANLEGEN --        <083>
12205 N=0:SA$="":A=0                           <072>
12210 IF II=0 THEN 12320                       <070>
12215 SYS PR,05,4,"SACHGEBIETE"                <115>
12216 SYS PR,05,5,"TTTTTTTTTTT"                <092>
12220 FOR I=1 TO II                            <209>
12230 M$(I)=SA$(I):SP(I)=5:ZE(I)=I+5           <147>
12240 NEXT                                     <056>
12245 M$(II+1)="NEUES SACHGEBIET ANLEGEN "
      :SP(II+1)=5:ZE(II+1)=18                  <184>
12246 M$(II+2)="SACHGEBIET LOESCHEN{6SPACE
      }":SP(II+2)=5:ZE(II+2)=19                <216>
12260 NN=II+2:GOSUB 14500                      <090>
12270 IF X$=CHR$(136) THEN A=136:RETURN        <114>
12280 IF J=II+1 THEN 12320                     <043>
12290 IF J=II+2 THEN 12700                     <056>
12310 SA$=SA$(J):GOSUB 10000:GOTO 12430        <143>
12320 GOSUB 11400:REM STATUSZEILE LOESCHEN     <189>
12322 IF II<10 THEN 12330                      <011>
12324 SYS PR,08,23,"{RVSON,SPACE}** NICHT
      MOEGLICH **{SPACE,RVOFF}":GOSUB 1110
      0:RETURN                                 <103>
12330 SYS PR,00,24,"NEUES SACHGEBIET EINGE
      BEN: "CHR$(145)                          <047>
12340 SYS IN,27,24,12,1,SA$                    <157>
12344 A=PEEK(828):IF A=136 THEN SA$="":RET
      URN                                      <011>
12345 IF A=145 OR A=17 OR SA$="{12SPACE}"T
      HEN 12340                                <198>
12350 FOR I=1 TO II                            <085>
12360 IF SA$=SA$(I) THEN SYS PR,00,23,"--
      NAME EXISTIERT SCHON --":GOTO 12330      <195>
```

```
12370 NEXT                                     <188>
12380 II=II+1:SA$(II)=SA$                      <134>
12390 GOSUB 10300:REM SACHGEBIETE WRITE        <081>
12420 GOSUB 10200:REM STATUSDATEI ANLEGEN      <004>
12430 RETURN                                   <040>
12440 :                                        <224>
12500 REM SACHGEBIETE WAEHLEN ---------        <108>
12510 N=0:SA$="":F=0:A=0                       <034>
12520 IF II=0 THEN F=1:RETURN                  <177>
12530 IF II=1 THEN SA$=SA$(1):RETURN           <038>
12534 SYS PR,05,4,"SACHGEBIETE"                <180>
12535 SYS PR,05,5,"TTTTTTTTTTT"                <157>
12536 FOR I=1 TO II                            <015>
12537 M$(I)=SA$(I):SP(I)=5:ZE(I)=I+5           <200>
12538 NEXT:NN=II                               <123>
12540 GOSUB 14500                              <040>
12545 IF X$=CHR$(136) THEN RETURN              <243>
12550 SA$=M$(J):RETURN                         <213>
12630 :                                        <160>
12700 REM SACHGEBIET LOESCHEN ---------        <011>
12710 KO$="{RVSON}SACHGEBIET LOESCHEN{RVOF
      F}":GOSUB 11000:GOSUB 12500              <095>
12720 IF A=136 THEN RETURN                     <070>
12725 A=J:IF II=1 THEN A=1                     <135>
12730 GOSUB 11400:SYS PR,1,23,"ZU LOESCHEN
      DES SACHGEBIET:"SA$(A)                   <151>
12740 SYS PR,10,24,"SICHER (J/N)?"CHR$(145
      )                                        <217>
12750 GET X$:IF X$<>"J"AND X$<>"N" THEN 12
      750                                      <112>
12760 IF X$="N" THEN A=136:RETURN              <045>
12765 GOSUB 11400:SYS PR,3,24,"... "SA$(A)
      " WIRD GELOESCHT ..."CHR$(145)           <052>
12770 IF A=II THEN 12810                       <246>
12780 FOR I=A TO II-1                          <034>
12790 SA$(I)=SA$(I+1)                          <112>
12800 NEXT                                     <110>
12810 II=II-1                                  <137>
12820 GOSUB 10300:REM  SACHGEBIETE WRITE       <003>
12830 OPEN 15,8,15,"S:"+SA$(A)+"*":CLOSE 1
      5                                        <232>
12831 OPEN 15,8,15,"S:??"+SA$(A)+"*":CLOSE
       15                                      <009>
12832 OPEN 15,8,15,"S:???"+SA$(A)+"*":CLOS
      E 15                                     <046>
12840 CLOSE 15:A=136:RETURN                    <002>
12850 :                                        <126>
13000 REM "AUFGABENDISKETTE EINLEGEN"          <193>
13010 SYS PR,6,23," AUFGABENDISKETTE EINLE
      GEN "                                    <030>
13020 FOR J=1 TO 80:NEXT                       <220>
13030 GET X$:IF X$<>""THEN 13070               <228>
13040 SYS PR,6,23,"{RVSON,SPACE}AUFGABENDI
      SKETTE EINLEGEN{SPACE,RVOFF}"            <066>
13050 FOR J=1 TO 80:NEXT                       <250>
13060 GOTO 13010                               <182>
13070 GOSUB 11400:RETURN                       <071>
13080 :                                        <102>
13200 REM AUSGABE RICHTIGE ANTWORT             <138>
13210 JJ=0:FOR J=0 TO LEN(AW$)                 <064>
13220 IF MID$(AW$,LEN(AW$)-J,1)<>" "THEN 1
      3240                                     <252>
13230 JJ=JJ+1:NEXT                             <153>
13240 AW$=LEFT$(AW$,LEN(AW$)-JJ)               <022>
13250 L=LEN(AW$)+19                            <102>
13270 RETURN                                   <118>
13280 :                                        <046>
14000 REM MENUE ***********************        <186>
14005 J=1                                      <242>
14010 GOSUB 14100:SYS PR,SP(J),ZE(J),"{RVS
      ON}"M$(J)CHR$(145)                       <017>
14015 GET X$                                   <181>
14017 IF X$=CHR$(13) THEN 14060                <034>
14020 IF X$=CHR$(133)OR X$=CHR$(29)OR X$=C
      HR$(157)OR X$=CHR$(136)THEN 14025        <072>
14022 IF X$<>CHR$(17)AND X$<>CHR$(145)THEN
       14015                                   <099>
14025 IF X$=CHR$(17)AND ZE(J)=23 AND J+3<=
      NN THEN J=J+3                            <218>
14026 IF X$=CHR$(145)AND ZE(J)=24 AND NN>4
       THEN J=J-3                              <012>
14030 IF X$=CHR$(133)OR X$=CHR$(136)THEN 1
      4060                                     <085>
14040 IF X$=CHR$(157)THEN J=J-1:IF J=0 THE
      N J=NN                                   <169>
```

```
14045 IF X$=CHR$(29)THEN J=J+1:IF J=NN+1 T
      HEN J=1                              <142>
14050 GOTO 14010                           <170>
14060 RETURN                               <146>
14070 :                                    <074>
14100 FOR I=1 TO NN                        <193>
14110 SYS PR,SP(I),ZE(I),M$(I)CHR$(145)    <044>
14120 NEXT:RETURN                          <089>
14500 REM MENUE ************************   <178>
14505 J=1                                  <234>
14510 GOSUB 14100:SYS PR,SP(J),ZE(J),"{RVS
      ON}"M$(J)CHR$(145)                   <009>
14515 GET X$                               <173>
14518 IF X$=CHR$(13) THEN 14560            <092>
14520 IF X$<>CHR$(133)AND X$<>CHR$(17)AND
      X$<>CHR$(145)AND X$<>CHR$(136)THEN 1
      4515                                 <155>
14530 IF X$=CHR$(133)OR X$=CHR$(136) THEN
      14560                                <087>
14540 IF X$=CHR$(145)THEN J=J-1:IF J=0 THE
      N J=NN                               <151>
14545 IF X$=CHR$(17)THEN J=J+1:IF J=NN+1 T
      HEN J=1                              <129>
14550 GOTO 14510                           <066>
14560 RETURN                               <138>
15000 REM DRUCKERANPASSUNG ************    <135>
15010 GOSUB 11400                          <146>
15020 M$(1)=" EPSON ":SP(1)=5:ZE(1)=24     <210>
15030 M$(2)=" MPS 801 ":SP(2)=23:ZE(2)=24:
      NN=2:GOSUB 14000                     <163>
15040 D=J:RETURN                           <104>
16000 REM PROGRAMMBESCHREIBUNG ********    <151>
16010 PRINT CHR$(147)                      <035>
16020 PRINT " MIT DIESEM PROGRAMM KANN MAN
      "                                    <197>
16030 PRINT " TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
      "                                    <014>
16040 PRINT "{2SPACE}- PRUEFUNGSFRAGEN LER
      NEN                                  <083>
16050 PRINT "{2SPACE}- PRUEFUNGSFRAGEN EIN
      GEBEN / AENDERN"                     <149>
16060 PRINT "{2SPACE}- PRUEFUNGSFRAGEN AUS
      DRUCKEN                              <147>
16070 PRINT                                <168>
16080 PRINT " ES IST BESONDERS FUER PROGRA
      MMIERTE{5SPACE}FRAGEN GEEIGNET.      <051>
16090 PRINT:PRINT                          <038>
16100 PRINT " FOLGENDENDE TASTEN WERDEN BE
      NUTZT:"                              <127>
16110 PRINT " TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
      TTTTTT":PRINT                        <189>
16120 PRINT " F1 / RETURN ZUR UEBERNAHME V
      ON WERTEN                            <198>
16130 PRINT "{13SPACE}UND MENUEPUNKTEN":PR
      INT                                  <081>
16140 PRINT" F7{10SPACE}ZUR RUECKKEHR ZUM
      VORHER-{15SPACE}GEHENDEN PRO";:      <070>
16150 PRINT "GRAMMTEIL":PRINT              <117>
16160 PRINT " CURSORSTEU- ZUR ANWAHL DER E
      INZELNEN"                            <009>
16170 PRINT " ERTASTEN{4SPACE}MENUEPUNKTE" <001>
16180 SYS PR,12,24,"{YELLOW}ZURUECK MIT F7
      {LIG.BLUE}"CHR$(145)                 <116>
16190 GET X$:IF X$<>CHR$(136)THEN 16190    <236>
16200 RETURN                               <000>
17000 REM KOPF -------------------------   <007>
17010 PRINT CHR$(147);:PRINT TAB(1)KO$     <035>
17015 IF SA$="" THEN 11020                 <043>
17016 SYS PR,1,1,SA$                       <038>
17017 SYS PR,28,0,"AUFG.NR.:"N-MAX+1       <254>
17018 SYS PR,28,1,"ANZAHL{2SPACE}:"N       <115>
17020 SYS PR,00,02,"TTTTTTTTTTTTTTTTTTTTTT
      TTTTTTTTTTTTTTTTTTT"CHR$(145)        <140>
17021 SYS PR,27,0,"T"                      <009>
17022 SYS PR,27,1,"T"                      <012>
17030 SYS PR,00,22,"TTTTTTTTTTTTTTTTTTTTTT
      TTTTTTTTTTTTTTTTTTT"CHR$(145)        <154>
17040 SYS PR,00,03,"";                     <203>
17050 RETURN                               <088>
17060 :                                    <016>
30000 OPEN 15,8,15                         <184>
30010 INPUT#15,A,A$                        <134>
30020 PRINT A,A$                           <145>
30030 CLOSE 15                             <150>
```


Listing 1. Das Programm Prüfungsfragen

```
programm : print obj    c000 c01a
----------------------------------

c000 : 20 fd ae 20 9e b7 86 d3  38
c008 : 20 fd ae 20 9e b7 a4 d3  b8
c010 : 18 20 f0 ff 20 fd ae 4c  ba
c018 : a4 aa 00 ff 00 ff 00 ff  11
```

Listing 2. PRINT. Diese Routine wird vom Hauptprogramm nachgeladen.

```
programm : in21 obj      c100 c22e
----------------------------------

c100 : 20 fd ae 20 9e b7 8e e9  84
c108 : 07 20 fd ae 20 9e b7 8e  67
c110 : eb 07 20 fd ae 20 9e b7  1c
c118 : 8e ec 07 ad ec 07 18 6d  d6
c120 : e9 07 8d ea 07 20 fd ae  14
c128 : 20 9e b7 8e ed 07 ac e9  f5
c130 : 07 ae eb 07 18 20 f0 ff  b0
c138 : a9 00 8d ee 07 a5 c6 85  e6
c140 : cc f0 fa 78 a5 cf f0 0c  07
c148 : a5 ce ae 87 02 a0 00 84  1f
c150 : cf 20 13 ea 20 b4 e5 8d  ac
c158 : 3c 03 c9 0d f0 18 c9 91  44
c160 : f0 14 c9 11 f0 74 c9 14  f1
c168 : f0 d3 c9 13 f0 cf c9 85  d6
c170 : 30 07 c9 8d 10 03 4c da  48
c178 : c1 c9 9d d0 11 c6 d3 a6  83
c180 : d3 ec e9 07 10 b7 ae e9  72
c188 : 07 86 d3 4c 3d c1 c9 1d  94
c190 : d0 0d ae ea 07 ca e4 d3  f2
c198 : f0 a3 e6 d3 4c 3d c1 c9  d7
c1a0 : 2e f0 28 c9 3a f0 96 c9  a3
c1a8 : 20 f0 20 a0 01 cc ed 07  99
c1b0 : 10 03 4c 08 af d0 03 4c  7c
c1b8 : cb c1 a0 0a a2 30 86 02  97
c1c0 : c5 02 f0 07 e8 88 d0 f6  a7
c1c8 : 4c 3d c1 20 d2 ff ae ea  e5
c1d0 : 07 e4 d3 d0 02 c6 d3 4c  96
c1d8 : 3d c1 ad e9 07 85 d3 20  cb
c1e0 : fd ae 20 8b b0 a6 0d f0  04
c1e8 : 42 85 fb 84 fc ad ec 07  7b
c1f0 : 20 7d b4 a2 02 a0 02 b5  e9
c1f8 : 61 91 fb ca 88 10 f8 a9  ba
c200 : 00 20 bd ff a9 03 a2 03  c3
c208 : a0 ff 20 ba ff 20 c0 ff  0b
c210 : a2 03 20 c6 ff a0 00 20  5a
c218 : cf ff 91 62 c8 cc ec 07  4c
c220 : d0 f5 20 cc ff a9 03 20  26
c228 : c3 ff 60 4c 99 ad 00 ff  93
```

Listing 3. Input-Routine

```
programm : bildschirm objc300 c375
----------------------------------

c300 : 20 fd ae 20 9e b7 8e e9  84
c308 : 07 20 fd ae 20 9e b7 8e  67
c310 : eb 07 20 fd ae 20 9e b7  1c
c318 : 8e ea 07 20 fd ae 20 9e  f4
c320 : b7 8e ec 07 ac e9 07 ae  ce
c328 : eb 07 18 20 f0 ff a6 d6  f8
c330 : ec eb 07 30 ef a5 c6 85  2c
c338 : cc f0 f3 78 a5 cf f0 0c  3d
c340 : a5 ce ae 87 02 a0 00 84  17
c348 : cf 20 13 ea 20 b4 e5 8d  a4
c350 : 3c 03 c9 85 f0 1e c9 88  69
c358 : f0 1a a6 d6 ec eb 07 10  44
c360 : 03 4c 24 c3 ec ec 07 30  bd
c368 : 05 c6 d6 4c 2e c3 20 d2  37
c370 : ff 4c 2e c3 60 ff 00 ff  9f
```

Listing 4. Texteingabe

```
programm : read10 obj   c500 c5ff
----------------------------------
c500 : a9 00 85 fa a9 50 85 fb  95
c508 : a2 02 20 c9 ff a9 78 85  27
c510 : fd a9 04 85 fe a2 12 a0  22
c518 : 00 b1 fd 85 fc 29 3f 06  43
c520 : fc 24 fc 10 02 09 80 70  bb
c528 : 02 09 40 91 fa c8 c0 28  3a
c530 : d0 e7 98 18 65 fd 85 fd  75
c538 : 90 02 e6 fe a9 0d 91 fa  a2
c540 : a5 fa 18 69 29 85 fa 90  61
c548 : 02 e6 fb ca d0 c9 ea a9  70
c550 : 00 85 fa a9 50 85 fb a2  6d
c558 : 12 a0 27 b1 fa c9 20 d0  da
c560 : 03 88 10 f7 a9 0d c8 91  f4
c568 : fa 18 a5 fa 69 29 85 fa  23
c570 : 90 02 e6 fb ca d0 e2 ea  cf
c578 : a9 b9 85 fa a9 52 85 fb  fa
c580 : a0 00 b1 fa c9 0d d0 10  54
c588 : 38 a5 fa e9 29 85 fa a5  85
c590 : fb e9 00 85 fb 4c 82 c5  e8
c598 : 18 a5 fa 69 29 85 fa 90  3a
c5a0 : 02 e6 fb a9 00 91 fa ea  98
c5a8 : a9 00 85 fa a9 50 85 fb  3d
c5b0 : a2 00 a0 00 b1 fa 20 d2  94
c5b8 : ff c9 00 f0 19 c9 0d f0  b0
c5c0 : 05 c8 c0 29 d0 ee 18 a5  af
c5c8 : fa 69 29 85 fa 90 02 e6  7c
c5d0 : fb e8 e0 12 d0 dc 20 cc  c8
c5d8 : ff 60 a2 02 20 c6 ff a2  6e
c5e0 : 00 a0 00 20 cf ff 20 d2  58
c5e8 : ff c9 0d f0 0c c9 00 f0  1e
c5f0 : 0d c8 c0 28 d0 ed 20 cf  33
c5f8 : ff e8 e0 12 d0 e3 60 a4  dd
```

Listing 5. Text lesen

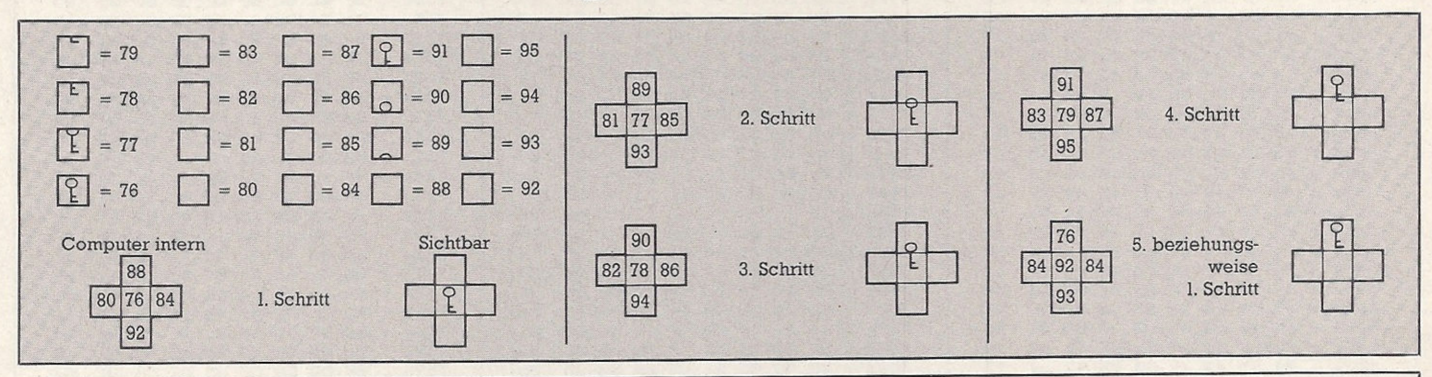
```
programm : drucker obj  c600 c622
----------------------------------
c600 : a9 01 a0 c6 a2 21 20 bd  5a
c608 : ff a9 01 a0 0f a6 02 20  9f
c610 : ba ff 20 c0 ff b0 02 a9  cb
c618 : 00 85 02 a9 01 20 c3 ff  b1
c620 : 60 20 00 ff 00 ff 00 ff  90
```

Listing 6. Drucker-Test

Der Schlüssel zum Zeichensatzscrolling

Ein Scrolling in vier Richtungen auf zwei Bildschirmhälften ist mit dem VIC eigentlich nicht möglich. Doch dieses Programm schafft es.

Um zu verdeutlichen, wie das Zeichensatzscrolling funktioniert, ein Beispiel. Nehmen wir hierfür einmal das beziehungsweise die Zeichen für den Schlüssel.




```
programm : schluessel        668a 79d2
--------------------------------------
668a : 20 18 e5 20 45 76 20 96  e9
6692 : 66 4c 93 66 20 b1 6c 78  02
669a : a9 bc 8d 14 03 a9 66 8d  b9
66a2 : 15 03 a9 30 8d 12 d0 ad  b1
66aa : 11 d0 29 7f 8d 11 d0 a9  56
66b2 : 81 8d 1a d0 a9 00 85 96  78
66ba : 58 60 ad 19 d0 8d 19 d0  50
66c2 : 29 01 d0 07 ad 0d dc 58  e8
66ca : 4c bc fe ad 12 d0 c9 fa  ae
66d2 : b0 28 c9 96 b0 12 ad 18  5e
66da : d0 29 f1 05 39 8d 18 d0  5e
66e2 : a9 96 8d 12 d0 4c bc fe  dc
66ea : ad 18 d0 29 f1 05 3b 8d  4c
66f2 : 18 d0 a9 fa 8d 12 d0 4c  81
66fa : bc fe ad 18 d0 29 f1 09  d4
6702 : 04 8d 18 d0 ad c2 04 c9  82
670a : b3 d0 07 a9 02 85 96 4c  5c
6712 : 1f 67 ad a2 06 c9 b3 d0  c4
671a : 04 a9 01 85 96 a5 96 f0  b7
6722 : 06 20 b2 6f 4c 46 68 4c  04
672a : 78 67 a5 3f 48 a5 40 48  ea
6732 : a5 41 48 a5 42 48 a2 00  30
673a : a0 00 b1 41 91 3f c8 c0  27
6742 : 0d d0 f7 e8 e0 0b f0 1d  37
674a : a5 3f 18 69 28 85 3f a5  b9
6752 : 40 69 00 85 40 a5 41 18  5e
675a : 69 28 85 41 a5 42 69 00  73
6762 : 85 42 4c 3a 67 a2 00 a0  30
676a : 00 68 85 42 68 85 41 68  d0
6772 : 85 40 68 85 3f 60 a5 3a  e4
677a : f0 18 c9 01 d0 03 4c 79  52
6782 : 68 c9 02 d0 03 4c ef 68  8d
678a : c9 03 d0 03 4c 65 69 4c  98
6792 : db 69 a9 36 85 3f a9 04  54
679a : 85 40 a5 43 85 41 a5 44  93
67a2 : 85 42 20 2c 67 ad 00 dc  74
67aa : c9 7e d0 07 a9 02 85 3a  fd
67b2 : 4c d3 67 c9 7d d0 07 a9  c9
67ba : 01 85 3a 4c d3 67 c9 7b  2d
67c2 : d0 07 a9 03 85 3a 4c d3  e4
67ca : 67 c9 77 d0 04 a9 04 85  b6
67d2 : 3a a5 3c f0 18 c9 01 d0  81
67da : 03 4c 7c 6a c9 02 d0 03  66
67e2 : 4c f2 6a c9 03 d0 03 4c  d6
67ea : 68 6b 4c de 6b a9 16 85  5e
67f2 : 3f a9 06 85 40 a5 45 85  89
67fa : 41 a5 46 85 42 20 2c 67  f5
6802 : ad 01 dc c9 fe d0 07 a9  86
680a : 02 85 3c 4c 2e 68 c9 fd  b1
6812 : d0 07 a9 01 85 3c 4c 2e  b8
681a : 68 c9 f7 d0 07 a9 04 85  58
6822 : 3c 4c 2e 68 c9 fb d0 04  e5
682a : a9 03 85 3c ad 1f d0 29  a7
6832 : 03 f0 05 85 96 4c 46 68  55
683a : 20 2d 6f ad 1e d0 29 03  95
6842 : f0 02 85 96 a9 30 8d 12  de
684a : d0 4c bc fe a2 00 fe 36  e2
6852 : 04 fe 5e 04 fe 86 04 fe  20
685a : ae 04 fe d6 04 fe fe 04  e1
6862 : fe 26 05 fe 4e 05 fe 76  8a
686a : 05 fe 9e 05 fe c6 05 e8  43
6872 : e0 0d d0 da a2 00 60 a9  67
687a : 08 85 39 a9 36 85 3f a9  a8
6882 : 04 85 40 a5 48 c9 e8 d0  26
688a : 07 a9 00 85 3a 4c a7 67  8a
6892 : a5 02 c9 03 d0 23 a5 43  4e
689a : 18 69 28 85 43 85 41 a5  d2
68a2 : 44 69 00 85 44 85 42 e6  93
68aa : 48 e6 48 a9 00 85 02 20  21
68b2 : 2c 67 a9 00 85 3a 4c a7  a7
68ba : 67 e6 02 ad 2c 05 c9 4c  75
68c2 : d0 21 ee c2 04 a9 00 8d  df
68ca : 2c 05 8d 04 05 8d 54 05  75
68d2 : 8d 2b 05 8d 2d 05 a5 43  00
68da : a4 44 85 4b 84 4c a2 01  a2
68e2 : 20 54 6c e6 48 e6 48 20  41
68ea : 4e 68 4c d3 67 a9 0a 85  f1
68f2 : 39 a9 36 85 3f a9 04 85  9a
68fa : 40 a5 48 c9 00 d0 07 a9  4e
6902 : 00 85 3a 4c a7 67 a5 02  2d
690a : c9 03 d0 23 a5 43 38 e9  16
6912 : 28 85 43 85 41 a5 44 e9  a4
691a : 00 85 44 85 42 c6 48 c6  a8
6922 : 48 a9 00 85 02 20 2c 67  90
692a : a9 00 85 3a 4c a7 67 e6  e9
6932 : 02 ad dc 04 c9 4c d0 21  47
693a : ee c2 04 a9 00 8d dc 04  a8
6942 : 8d 04 05 8d b4 04 8d db  1e
694a : 04 8d dd 04 a5 43 a4 44  9c
6952 : 85 4b 84 4c a2 02 20 54  8b
695a : 6c c6 48 c6 48 20 4e 68  a4
6962 : 4c d3 67 a9 0c 85 39 a9  cc
696a : 36 85 3f a9 04 85 40 a5  21
6972 : 47 c9 00 d0 07 a9 00 85  81
697a : 3a 4c a7 67 a5 02 c9 03  49
6982 : d0 23 a5 43 38 e9 01 85  97
698a : 43 85 41 a5 44 e9 00 85  33
6992 : 44 85 42 c6 47 c6 47 a9  1d
699a : 00 85 02 20 2c 67 a9 00  86
69a2 : 85 3a 4c a7 67 e6 02 ad  5d
69aa : 03 05 c9 4c d0 21 ee c2  83
69b2 : 04 a9 00 8d 03 05 8d 04  d3
69ba : 05 8d 02 05 8d db 04 8d  8a
69c2 : 2b 05 a5 43 a4 44 85 4b  5b
69ca : 84 4c a2 03 20 54 6c c6  61
69d2 : 47 c6 47 20 4e 68 4c d3  53
69da : 67 a9 0e 85 39 a9 36 85  0f
69e2 : 3f a9 04 85 40 a5 47 c9  89
69ea : d8 d0 07 a9 00 85 3a 4c  cf
69f2 : a7 67 a5 02 c9 03 d0 23  35
69fa : a5 43 18 69 01 85 43 85  c8
6a02 : 41 a5 44 69 00 85 44 85  9c
6a0a : 42 e6 47 e6 47 a9 00 85  3b
6a12 : 02 20 2c 67 a9 00 85 3a  41
6a1a : 4c a7 67 e6 02 ad 05 05  9c
6a22 : c9 4c d0 21 ee c2 04 a9  d2
6a2a : 00 8d 05 05 8d 06 05 8d  0b
6a32 : 04 05 8d dd 04 8d 2d 05  43
6a3a : a5 43 a4 44 85 4b 84 4c  90
6a42 : a2 04 20 54 6c e6 47 e6  62
6a4a : 47 20 4e 68 4c d3 67 a2  88
6a52 : 00 fe 16 06 fe 3e 06 fe  10
6a5a : 66 06 fe 8e 06 fe b6 06  94
6a62 : fe de 06 fe 06 07 fe 2e  22
6a6a : 07 fe 56 07 fe 7e 07 fe  65
6a72 : a6 07 e8 e0 0d d0 da a2  fa
6a7a : 00 60 a9 08 85 3b a9 16  1b
6a82 : 85 3f a9 06 85 40 a5 4a  57
6a8a : c9 e8 d0 07 a9 00 85 3c  05
6a92 : 4c 02 68 a5 0c c9 03 d0  6b
6a9a : 23 a5 45 18 69 28 85 45  5d
6aa2 : 85 41 a5 46 69 00 85 46  33
6aaa : 85 42 e6 4a e6 4a a9 00  bb
6ab2 : 85 0c 20 2c 67 a9 00 85  9a
6aba : 3c 4c 02 68 e6 0c ad 0c  48
6ac2 : 07 c9 4c d0 21 ee a2 06  fb
6aca : a9 00 8d 0c 07 8d e4 06  d5
6ad2 : 8d 34 07 8d 0b 07 8d 0d  26
6ada : 07 a5 45 a4 46 85 4b 84  60
6ae2 : 4c a2 01 20 54 6c e6 4a  9d
6aea : e6 4a 20 51 6a 4c 2e 68  ba
6af2 : a9 0a 85 3b a9 16 85 3f  49
6afa : a9 06 85 40 a5 4a c9 00  e4
6b02 : d0 07 a9 00 85 3c 4c 02  30
6b0a : 68 a5 0c c9 03 d0 23 a5  10
6b12 : 45 38 e9 28 85 45 85 41  0e
6b1a : a5 46 e9 00 85 46 85 42  82
6b22 : c6 4a c6 4a a9 00 85 0c  d1
6b2a : 20 2c 67 a9 00 85 3c 4c  25
6b32 : 02 68 e6 0c ad bc 06 c9  10
6b3a : 4c d0 21 ee a2 06 a9 00  15
6b42 : 8d bc 06 8d e4 06 8d 94  3e
6b4a : 06 8d bb 06 8d bd 06 a5  f1
6b52 : 45 a4 46 85 4b 84 4c a2  7b
6b5a : 02 20 54 6c c6 4a c6 4a  7d
6b62 : 20 51 6a 4c 2e 68 a9 0c  34
6b6a : 85 3b a9 16 85 3f a9 06  bf
6b72 : 85 40 a5 49 c9 00 d0 07  98
6b7a : a9 00 85 3c 4c 02 68 a5  ce
6b82 : 0c c9 03 d0 23 a5 45 38  33
6b8a : e9 01 85 45 85 41 a5 46  83
6b92 : e9 00 85 46 85 42 c6 49  be
6b9a : c6 49 a9 00 85 0c 20 2c  01
6ba2 : 67 a9 00 85 3c 4c 02 68  8e
6baa : e6 0c ad e3 06 c9 4c d0  00
6bb2 : 21 ee a2 06 a9 00 8d e3  4c
6bba : 06 8d e4 06 8d e2 06 8d  a4
6bc2 : bb 06 8d 0b 07 a5 45 a4  41
6bca : 46 85 4b 84 4c a2 03 20  5c
6bd2 : 54 6c c6 49 c6 49 20 51  11
6bda : 6a 4c 2e 68 a9 0e 85 3b  9b
6be2 : a9 16 85 3f a9 06 85 40  41
6bea : a5 49 c9 d8 d0 07 a9 00  ad
6bf2 : 85 3c 4c 02 68 a5 0c c9  60
6bfa : 03 d0 23 a5 45 18 69 01  a0
6c02 : 85 45 85 41 a5 46 69 00  e6
6c0a : 85 46 85 42 e6 49 e6 49  43
6c12 : a9 00 85 0c 20 2c 67 a9  f3
6c1a : 00 85 3c 4c 02 68 e6 0c  8d
6c22 : ad e5 06 c9 4c d0 21 ee  2a
6c2a : a2 06 a9 00 8d e5 06 8d  75
6c32 : e6 06 8d e4 06 8d bd 06  eb
6c3a : 8d 0d 07 a5 45 a4 46 85  62
6c42 : 4b 84 4c a2 04 20 54 6c  a2
6c4a : e6 49 e6 49 20 51 6a 4c  86
6c52 : 2e 68 a5 4b 18 69 7e 85  59
6c5a : 4b a5 4c 69 00 85 4c a9  69
6c62 : 00 a0 50 91 4b e0 01 d0  5a
6c6a : 0f a0 77 91 4b c8 91 4b  b1
6c72 : c8 91 4b a0 a0 91 4b 60  6e
```

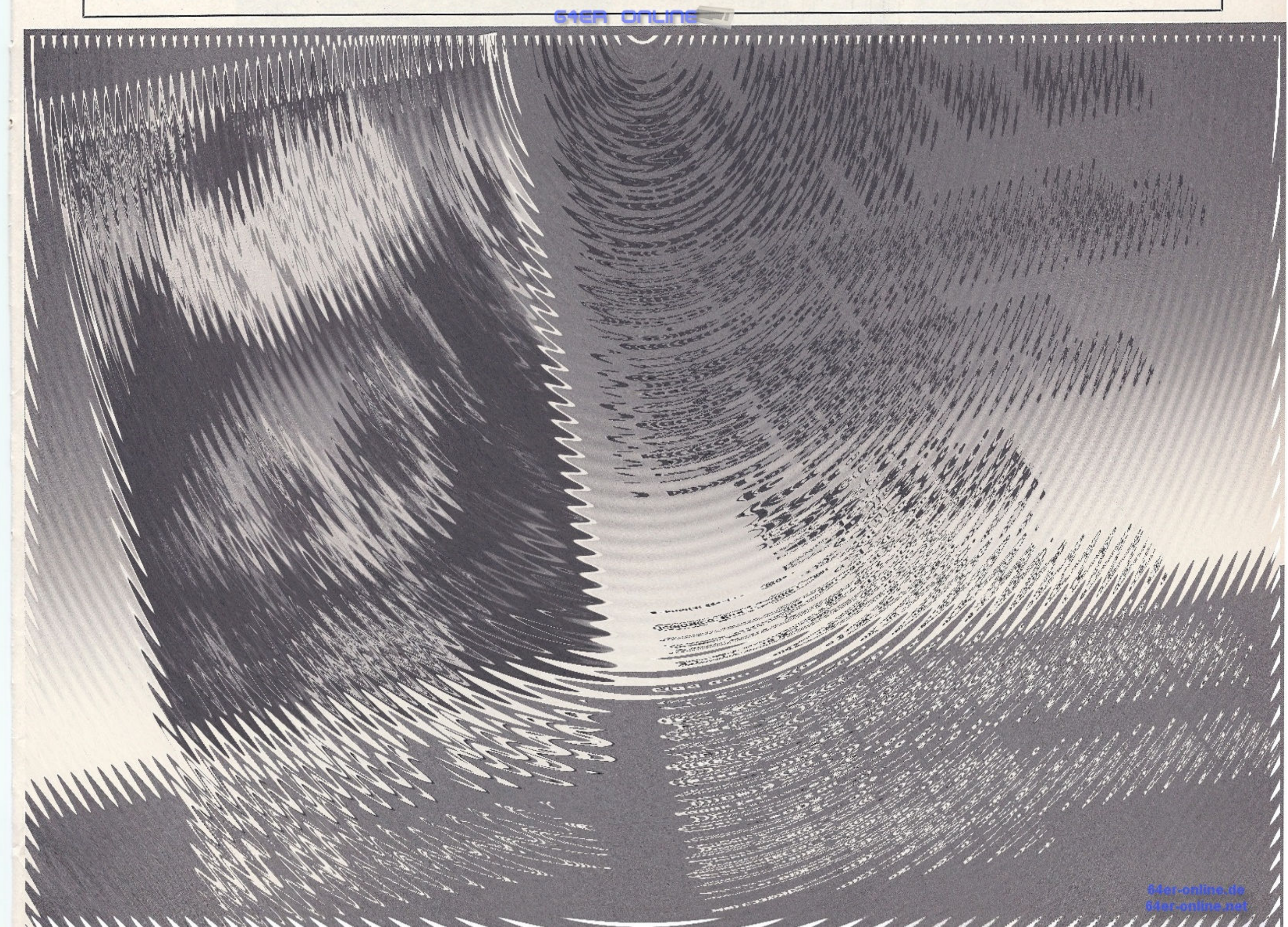
Listing »Schlüssel«. Bitte beachten Sie bei der Eingabe den MSE.

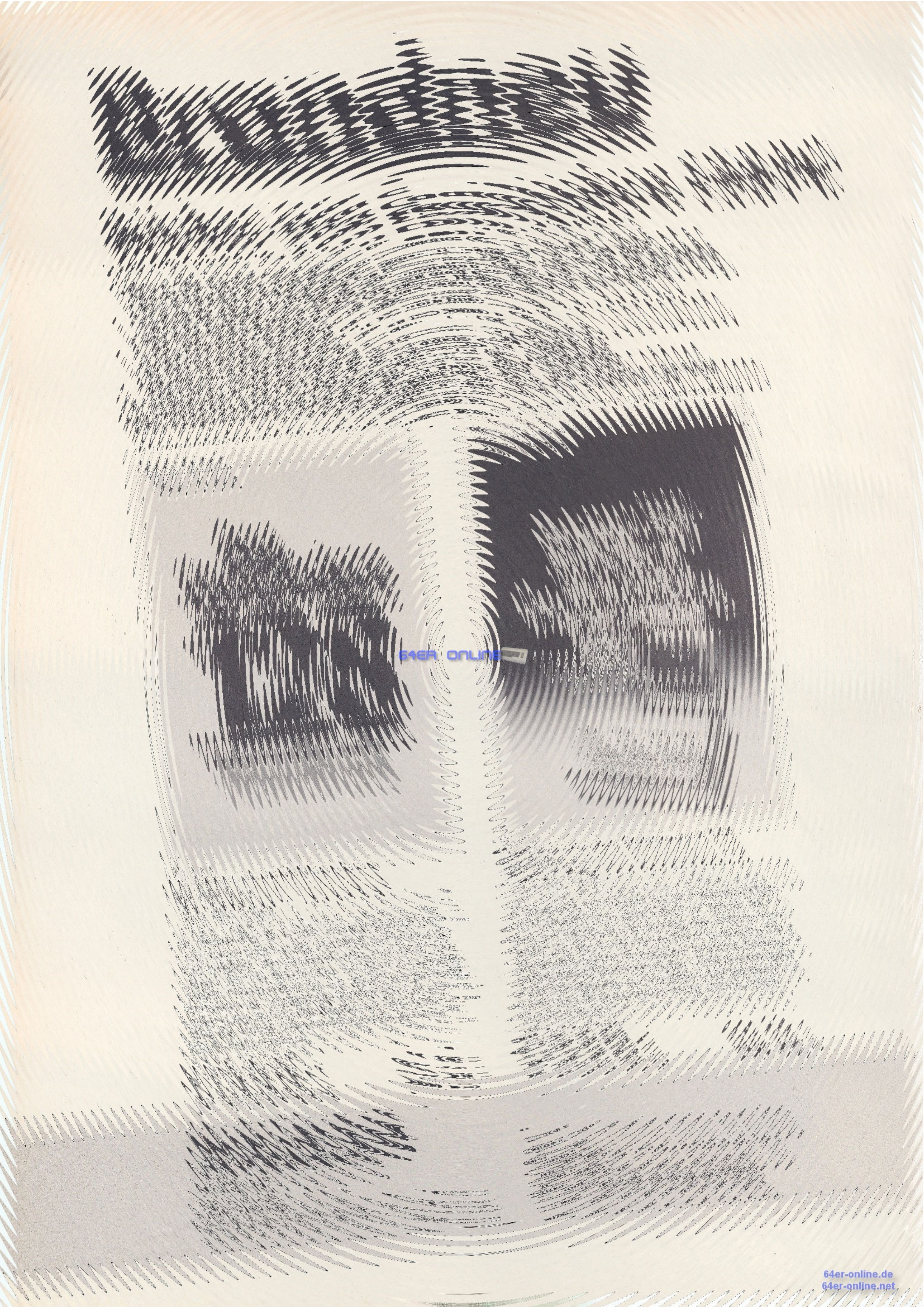
```
6c7a : e0 02 d0 0f a0 00 91 4b   58
6c82 : a0 27 91 4b c8 91 4b c8   5b
6c8a : 91 4b 60 e0 03 d0 10 a0   2d
6c92 : 27 91 4b a0 77 91 4b a0   db
6c9a : 4e 91 4b c8 91 4b 60 a0   d3
6ca2 : 29 91 4b a0 79 91 4b a0   0d
6caa : 51 91 4b c8 91 4b 60 20   e5
6cb2 : 44 e5 a0 00 a9 00 99 00   12
6cba : d0 c8 c0 11 d0 f8 8d 15   76
6cc2 : d0 a9 07 8d 20 d0 a9 00   0a
6cca : 8d 21 d0 a0 00 a9 07 99   cd
6cd2 : 00 d8 99 c8 d8 99 90 d9   0e
6cda : 99 58 da 99 20 db c8 c0   0f
6ce2 : c8 d0 ec a9 0d 85 4b a9   00
6cea : d8 85 4c a0 00 a9 05 91   30
6cf2 : 4b c8 c0 0f d0 f9 a0 00   13
6cfa : a2 00 a9 05 91 4b a5 4b   48
6d02 : 18 69 28 85 4b a5 4c 69   6f
6d0a : 00 85 4c e8 e0 19 d0 ea   ed
6d12 : a9 1b 85 4b a9 d8 85 4c   24
6d1a : a2 00 a9 05 91 4b a9 05   eb
6d22 : a5 4b 18 69 28 85 4b a5   c7
6d2a : 4c 69 00 85 4c e8 e0 19   9d
6d32 : d0 e8 a9 ed 85 4b a9 d9   ab
6d3a : 85 4c a0 00 a9 05 91 4b   ad
6d42 : c8 c0 0f d0 f9 a0 00 a9   40
6d4a : cd 85 4b a9 db 85 4c a9   50
6d52 : 05 91 4b c8 c0 0f d0 f9   c7
6d5a : a9 00 a0 70 85 5f 84 60   60
6d62 : a9 41 a0 76 85 5a 84 5b   97
6d6a : a9 41 a0 86 85 58 84 59   8d
6d72 : 20 bf a3 a9 36 85 3f a9   70
6d7a : 04 85 40 a9 80 85 42 a9   17
6d82 : 00 85 41 20 2c 67 a9 16   6a
6d8a : 85 3f a9 06 85 40 a9 1b   11
6d92 : 85 41 a9 80 85 42 20 2c   75
6d9a : 67 a0 00 b9 ab 6d 99 c0   97
6da2 : 02 c8 c0 3f d0 f5 4c ea   e4
6daa : 6d ff 00 00 e7 00 00 c3   1d
6db2 : 00 00 c3 00 00 e7 00 00   e2
6dba : e7 00 00 e7 00 00 ff 00   9e
6dc2 : 00 00 00 00 00 00 00 00   c3
6dca : 00 00 00 00 00 00 00 00   cb
6dd2 : 00 00 00 00 00 00 00 00   d3
6dda : 00 00 00 00 00 00 00 00   db
6de2 : 00 00 00 00 00 00 00 00   e3
6dea : a9 0b 8d f8 07 8d f9 07   6e
6df2 : 8d fa 07 8d fb 07 a9 0e   2b
6dfa : 8d 27 d0 8d 29 d0 a9 0a   d4
6e02 : 8d 28 d0 8d 2a d0 a9 03   5f
6e0a : 8d 15 d0 a9 b8 8d 00 d0   25
6e12 : 8d 02 d0 a9 62 8d 01 d0   42
6e1a : a9 c2 8d 03 d0 a0 00 b9   6e
6e22 : 54 6e 99 53 04 99 33 06   64
6e2a : c8 c0 06 d0 f2 a9 b1 8d   4c
6e32 : 59 04 a9 b2 8d 39 06 a0   4a
6e3a : 00 b9 5a 6e 99 6d 04 99   c3
6e42 : 4d 06 c8 c0 0a d0 f2 a9   23
6e4a : b0 8d c2 04 8d a2 06 4c   91
6e52 : 64 6e 90 8c 81 99 85 92   c3
6e5a : 93 83 88 8c 95 85 93 93   5d
6e62 : 85 8c a9 00 85 43 a9 80   b2
6e6a : 85 44 a9 00 85 47 85 48   b5
6e72 : 85 4a 85 3d 85 3e a9 d8   c8
6e7a : 85 49 a9 1b 85 45 a9 80   9c
6e82 : 85 46 a9 08 85 39 85 3b   44
6e8a : a9 00 85 02 85 0c 85 3a   18
6e92 : 85 3c ad 1f d0 ad 1e d0   19
6e9a : 60 a5 b3 c9 01 d0 46 a5   ee
6ea2 : ac 38 e5 47 85 ac a5 ad   7c
6eaa : e9 00 85 ad a5 ae 38 e5   27
6eb2 : 48 85 ae a5 af e9 00 85   72
6eba : af a5 ad d0 23 a5 af d0   81
6ec2 : 1f a5 ac c9 61 b0 19 a5   63
6eca : ae c9 51 b0 13 a9 01 85   55
6ed2 : b2 a5 ac 18 69 88 85 b0   d7
6eda : a5 ae 18 69 3a 85 b1 60   61
6ee2 : a9 00 85 b2 60 a5 ac 38   99
6eea : e5 49 85 ac a5 ad e9 00   da
6ef2 : 85 ad a5 ae 38 e5 4a 85   74
6efa : ae a5 af e9 00 85 af a5   da
6f02 : ad d0 23 a5 af d0 1f a5   de
6f0a : ac c9 61 b0 19 a5 ae c9   16
6f12 : 51 b0 13 a9 01 85 b2 a5   08
6f1a : ac 18 69 88 85 b0 a5 ae   10
6f22 : 18 69 9a 85 b1 60 a9 00   0b
6f2a : 85 b2 60 a5 47 18 69 30   10
6f32 : 85 ac a9 00 69 00 85 ad   80
6f3a : a5 48 18 69 28 85 ae a9   f3
6f42 : 00 69 00 85 af a9 02 85   03
6f4a : b3 20 9b 6e a5 b2 f0 15   a0
6f52 : ad 15 d0 09 04 8d 15 d0   82
6f5a : a5 b0 8d 04 d0 a5 b1 8d   57
6f62 : 05 d0 4c 6f 6f ad 15 d0   2b
6f6a : 29 f6 8d 15 d0 a5 49 18   26
6f72 : 69 30 85 ac a9 00 69 00   2a
6f7a : 85 ad a5 4a 18 69 28 85   01
6f82 : ae a9 00 69 00 85 af a9   70
6f8a : 01 85 b3 20 9b 6e a5 b2   68
6f92 : f0 15 ad 15 d0 09 08 8d   ab
6f9a : 15 d0 a5 b0 8d 06 d0 a5   2e
6fa2 : b1 8d 07 d0 4c b1 6f ad   61
6faa : 15 d0 29 f7 8d 15 d0 60   f6
6fb2 : a5 96 c9 03 f0 04 c9 01   cd
6fba : d0 16 a0 00 b9 e2 6f 99   61
6fc2 : a3 04 c8 c0 04 d0 f5 a5   9b
6fca : 96 c9 03 f0 03 4c e6 6f   31
6fd2 : a0 00 b9 e2 6f 99 83 06   1b
6fda : c8 c0 04 d0 f5 4c e6 6f   5a
6fe2 : 8c 8f 93 94 ad 00 dc c9   8f
6fea : 6f f0 01 60 20 b1 6c a9   b2
6ff2 : 00 85 96 60 00 00 00 00   67
6ffa : 00 00 00 00 00 00 28 04   a3
7002 : 04 04 04 04 04 04 04 04   02
700a : 04 04 04 04 04 04 04 04   0a
7012 : 04 04 04 04 04 04 04 04   12
701a : 04 04 04 04 04 04 04 04   1a
7022 : 04 04 04 04 04 30 10 38   1c
702a : 0c 0c 0c 0c 0c 0c 0c 0c   2a
7032 : 0c 0c 0c 0c 0c 0c 0c 0c   32
703a : 0c 0c 0c 0c 0c 0c 0c 0c   3a
7042 : 0c 0c 0c 0c 0c 0c 0c 0c   42
704a : 0c 0c 0c 0c 40 10 10 48   36
7052 : 00 08 00 00 00 08 00 00   97
705a : 00 00 00 00 00 00 00 00   5b
7062 : 00 00 00 00 00 00 00 00   63
706a : 00 00 00 00 00 00 08 00   8b
7072 : 00 08 00 00 44 10 10 48   0c
707a : 44 14 48 00 44 14 48 00   e1
7082 : 00 00 00 00 00 00 00 00   83
708a : 00 00 00 00 00 00 00 00   8b
7092 : 00 00 00 00 00 44 14 48   96
709a : 44 14 48 00 44 10 10 48   90
70a2 : 44 18 48 00 44 10 48 00   ea
70aa : 00 00 00 00 00 00 00 00   ab
70b2 : 00 00 00 00 00 00 00 00   b3
70ba : 00 00 00 00 00 44 10 48   ad
70c2 : 44 10 48 00 44 10 10 48   b6
70ca : 00 0c 00 00 44 10 48 00   b6
70d2 : 08 08 08 08 08 08 08 08   d2
70da : 08 08 08 08 08 08 08 08   da
70e2 : 08 08 08 08 00 44 10 48   e4
70ea : 44 10 48 00 44 10 10 48   de
70f2 : 00 08 00 00 44 10 48 44   65
70fa : 28 04 04 04 04 04 04 04   1e
7102 : 04 04 04 04 04 04 04 04   02
710a : 04 04 04 30 48 44 10 48   8f
7112 : 44 10 48 00 44 10 10 48   06
711a : 44 14 48 00 44 10 48 44   e9
7122 : 10 38 0c 0c 0c 0c 0c 0c   3c
712a : 0c 0c 0c 0c 0c 0c 0c 0c   2a
7132 : 0c 0c 40 10 48 44 10 48   ce
713a : 44 18 48 00 44 10 10 48   32
7142 : 44 18 48 00 44 10 48 44   13
714a : 10 48 00 58 00 00 08 08   ba
7152 : 08 08 08 08 08 08 00 00   22
715a : 08 00 44 10 48 44 10 48   ed
7162 : 00 0c 00 00 44 10 10 48   fe
716a : 00 0c 00 00 44 10 48 44   df
7172 : 10 48 50 4c 54 44 1c 04   24
717a : 04 04 04 04 04 20 48 44   ed
7182 : 14 48 44 10 48 44 10 48   45
718a : 00 08 00 00 44 10 10 48   24
7192 : 00 08 00 00 44 10 48 44   05
719a : 10 48 00 5c 00 00 0c 0c   a2
71a2 : 0c 0c 0c 0c 0c 0c 00 44   e3
71aa : 10 48 44 10 48 44 10 48   69
71b2 : 44 14 48 00 44 10 10 48   a8
71ba : 44 14 48 00 44 10 48 44   89
71c2 : 10 34 08 08 08 08 08 08   e0
71ca : 08 08 08 08 08 08 08 3c   33
71d2 : 10 48 44 10 48 44 10 48   91
71da : 44 18 48 00 44 10 10 48   d2
71e2 : 44 10 48 00 44 10 48 44   af
71ea : 24 04 04 04 04 04 04 04   0a
71f2 : 04 04 04 04 04 04 04 04   f2
71fa : 2c 48 44 10 48 44 10 48   d5
7202 : 00 0c 00 00 44 10 10 48   9e
720a : 44 10 48 00 44 10 48 00   4e
7212 : 0c 0c 0c 0c 0c 0c 0c 0c   12
721a : 0c 0c 0c 0c 0c 0c 0c 0c   1a
7222 : 0c 00 44 10 48 44 10 48   b9
722a : 00 08 00 00 44 10 10 48   c4
7232 : 44 10 48 00 44 10 34 08   36
723a : 08 08 08 08 08 08 08 08   3a
7242 : 08 08 08 08 08 08 08 08   42
724a : 08 08 3c 10 48 44 10 48   df
7252 : 44 14 48 00 44 10 10 48   48
725a : 44 18 48 00 44 24 04 04   3a
7262 : 04 04 04 04 04 04 04 04   62
726a : 04 04 04 04 04 04 04 04   6a
7272 : 04 04 04 2c 48 44 10 48   76
727a : 44 10 48 00 44 10 10 48   6e
7282 : 00 0c 00 00 00 0c 0c 0c   31
728a : 0c 0c 0c 0c 0c 0c 0c 0c   8a
7292 : 0c 0c 0c 0c 0c 0c 0c 0c   92
729a : 0c 0c 0c 0c 00 44 10 48   24
72a2 : 44 10 48 00 44 10 10 48   96
72aa : 00 08 08 08 00 00 08 08   e2
72b2 : 08 08 08 08 08 08 08 08   b2
72ba : 08 08 08 08 00 00 08 08   fa
72c2 : 08 08 08 08 08 3c 10 48   05
72ca : 44 10 48 00 44 10 10 48   be
72d2 : 44 1c 04 20 48 44 1c 04   49
72da : 04 04 04 04 04 04 04 04   da
72e2 : 04 04 04 20 48 44 1c 04   0d
72ea : 04 04 04 04 04 04 2c 48   14
72f2 : 44 18 48 00 44 10 10 48   ea
72fa : 00 0c 0c 0c 00 00 0c 0c   cd
7302 : 0c 0c 0c 0c 0c 0c 0c 0c   02
730a : 0c 0c 0c 0c 00 00 0c 0c   e9
7312 : 0c 0c 0c 0c 0c 0c 0c 00   fa
731a : 00 0c 00 00 44 10 10 48   b6
7322 : 00 08 08 08 08 08 08 08   1a
732a : 00 00 00 00 00 00 00 00   2b
7332 : 08 08 08 08 00 00 00 00   42
733a : 00 00 00 00 08 08 08 08   2b
7342 : 08 08 08 00 44 10 10 48   e6
734a : 44 1c 04 04 04 04 04 20   cf
7352 : 48 08 08 08 08 08 08 44   0b
735a : 1c 04 04 20 48 08 08 08   72
7362 : 08 08 08 44 1c 04 04 04   f3
736a : 04 04 20 48 44 10 10 48   17
7372 : 00 0c 0c 0c 0c 0c 0c 0c   66
737a : 44 1c 04 04 04 04 20 48   bf
7382 : 0c 0c 0c 0c 44 1c 04 04   56
738a : 04 04 20 48 0c 0c 0c 0c   0b
7392 : 0c 0c 0c 00 44 10 10 48   3d
739a : 00 00 08 08 08 08 08 08   8e
73a2 : 08 0c 0c 0c 0c 0c 0c 08   96
73aa : 08 08 08 08 08 0c 0c 0c   e3
73b2 : 0c 0c 0c 08 08 08 08 08   b9
73ba : 08 08 00 00 44 10 10 48   5c
73c2 : 00 44 1c 04 04 04 04 04   e4
73ca : 20 48 08 08 08 08 44 1c   1c
73d2 : 04 04 04 04 20 48 08 08   ce
73da : 08 08 44 1c 04 04 04 04   f3
73e2 : 04 20 48 00 44 10 10 48   9e
73ea : 00 08 0c 0c 0c 0c 0c 0c   dc
73f2 : 0c 44 1c 04 04 20 48 0c   22
73fa : 0c 0c 0c 0c 0c 44 1c 04   ec
7402 : 04 20 48 0c 0c 0c 0c 0c   13
740a : 0c 0c 08 00 44 10 10 48   b4
7412 : 44 14 48 00 00 08 08 08   e3
741a : 08 08 0c 0c 0c 0c 08 08   fc
7422 : 08 08 08 08 08 08 0c 0c   3b
742a : 0c 0c 08 08 08 08 08 00   20
7432 : 00 44 14 48 44 10 10 48   f8
743a : 44 18 48 00 44 1c 04 04   da
7442 : 04 20 48 08 08 44 1c 04   85
744a : 04 04 04 04 04 20 48 08   44
7452 : 08 44 1c 04 04 04 20 48   75
745a : 00 44 18 48 44 10 10 48   21
7462 : 00 0c 00 00 00 0c 0c 0c   11
746a : 0c 0c 44 1c 20 48 0c 0c   9e
7472 : 0c 0c 0c 0c 0c 0c 44 1c   73
747a : 20 48 0c 0c 0c 0c 0c 00   94
7482 : 00 00 0c 00 44 10 10 48   1b
748a : 00 00 00 00 00 08 08 08   fb
7492 : 08 08 08 0c 0c 00 00 58   93
749a : 00 00 00 00 58 00 00 0c   38
74a2 : 0c 08 08 08 08 08 08 00   96
74aa : 00 00 00 00 44 10 10 48   40
74b2 : 00 00 00 00 44 28 04 04   50
74ba : 04 04 30 48 00 00 50 4c   af
74c2 : 54 00 00 50 4c 54 00 00   88
74ca : 44 28 04 04 04 04 30 48   56
74d2 : 00 00 00 00 44 10 10 48   68
74da : 00 00 00 00 44 10 38 0c   98
74e2 : 0c 40 10 48 00 00 00 5c   d4
74ea : 00 00 00 00 5c 00 00 00   b0
74f2 : 44 10 38 0c 0c 40 10 48   62
74fa : 00 00 00 00 44 10 10 48   90
7502 : 00 00 00 00 44 18 48 58   d9
750a : 00 44 10 34 08 08 08 08   a8
7512 : 08 08 08 08 08 08 08 08   12
751a : 3c 10 48 00 58 44 18 48   09
7522 : 00 00 00 00 44 10 10 48   b8
752a : 00 00 00 00 00 0c 50 4c   65
7532 : 54 44 24 04 04 04 04 04   aa
753a : 04 04 04 04 04 04 04 04   3a
7542 : 04 2c 48 50 4c 54 0c 00   10
754a : 00 00 00 00 44 10 10 48   e0
7552 : 00 00 00 00 00 00 00 5c   0b
755a : 00 00 0c 0c 0c 0c 0c 0c   48
7562 : 0c 0c 0c 0c 0c 0c 0c 0c   62
756a : 0c 0c 00 00 5c 00 00 00   42
7572 : 00 00 00 00 44 10 10 48   08
757a : 00 00 00 00 00 00 00 00   7b
7582 : 00 00 00 00 00 00 00 00   83
758a : 00 00 00 00 00 00 00 00   8b
7592 : 00 00 00 00 00 00 00 00   93
```

Listing »Schlüssel« (Fortsetzung)

```
759a : 00 00 00 00 44 10 10 48  30
75a2 : 00 00 00 00 00 00 00 00  a3
75aa : 00 00 00 00 00 00 00 00  ab
75b2 : 00 00 00 00 00 00 00 00  b3
75ba : 00 00 00 00 00 00 00 00  bb
75c2 : 00 00 00 00 44 10 10 48  58
75ca : 00 00 00 00 00 00 00 00  cb
75d2 : 00 00 00 00 00 00 00 00  d3
75da : 00 00 00 00 00 00 00 00  db
75e2 : 00 00 00 00 00 00 00 00  e3
75ea : 00 00 00 00 44 10 10 34  58
75f2 : 08 08 08 08 08 08 08 08  f2
75fa : 08 08 08 08 08 08 08 08  fa
7602 : 08 08 08 08 08 08 08 08  02
760a : 08 08 08 08 08 08 08 08  0a
7612 : 08 08 08 08 3c 10 24 04  fe
761a : 04 04 04 04 04 04 04 04  1a
7622 : 04 04 04 04 04 04 04 04  22
762a : 04 04 04 04 04 04 04 04  2a
7632 : 04 04 04 04 04 04 04 04  32
763a : 04 04 04 04 04 2c da 07  dd
7642 : 84 36 30 a9 20 85 42 a9  ad
764a : 00 85 41 a0 00 a9 00 91  e2
7652 : 41 a5 41 18 69 01 85 41  f0
765a : a5 42 69 00 85 42 a6 42  04
7662 : e0 40 d0 e9 a9 ff 99 80  d6
766a : 20 99 a0 20 99 40 21 99  d6
7672 : 80 21 99 80 28 99 c0 28  9c
767a : 99 20 29 99 60 29 99 20  97
7682 : 30 99 00 31 99 60 31 99  3a
768a : 80 31 99 20 38 99 e0 38  51
7692 : 99 20 39 99 40 39 c8 c0  2f
769a : 20 d0 cb a0 00 b9 d2 77  31
76a2 : 99 20 20 99 c0 20 99 e0  bc
76aa : 20 99 00 21 99 20 21 99  0d
76b2 : 60 21 c8 c0 20 d0 e6 a0  52
76ba : 00 b9 f2 77 99 40 20 99  92
76c2 : a0 21 99 e0 21 c8 c0 20  11
76ca : d0 ef a0 00 b9 52 78 99  fd
76d2 : 20 28 99 a0 28 99 e0 28  a4
76da : 99 00 29 99 40 29 99 80  a6
76e2 : 29 c8 c0 20 d0 e6 a0 00  6a
76ea : b9 72 78 99 60 28 99 c0  5d
76f2 : 29 99 00 2a c8 c0 20 d0  e2
76fa : ef a0 00 b9 d2 78 99 80  c9
7702 : 30 99 a0 30 99 c0 30 99  c1
770a : e0 30 99 20 31 99 40 31  b0
7712 : c8 c0 20 d0 e6 a0 00 b9  43
771a : f2 78 99 a0 31 99 c0 31  08
7722 : 99 40 32 c8 c0 20 d0 ef  b1
772a : a0 00 b9 52 79 99 80 38  5a
7732 : 99 a0 38 99 c0 38 99 00  91
773a : 39 99 60 39 99 80 39 c8  93
7742 : c0 20 d0 e6 a0 00 b9 72  f9
774a : 79 99 e0 39 99 00 3a 99  a5
7752 : 20 3a c8 c0 20 d0 ef ad  7d
775a : 0e dc 29 fe 8d 0e dc a5  08
7762 : 01 29 fb 85 01 a0 00 b9  30
776a : 00 d4 99 00 24 99 00 2c  a2
7772 : 99 00 34 99 00 3c b9 00  14
777a : d5 99 00 25 99 00 2d 99  42
7782 : 00 35 99 00 3d c8 c0 ff  a0
778a : d0 dd a5 01 09 04 85 01  9b
7792 : ad 0e dc 09 01 8d 0e dc  0d
779a : a0 00 b9 12 78 99 60 22  05
77a2 : b9 32 78 99 c0 22 b9 92  ef
77aa : 78 99 60 2a b9 b2 78 99  92
77b2 : e0 2a b9 12 79 99 60 32  a2
77ba : b9 32 79 99 a0 32 b9 92  c5
77c2 : 79 99 60 3a b9 b2 79 99  b1
77ca : 80 3a c8 c0 20 d0 cb 60  2a
77d2 : ff ff ff ff ff ff ff ff  d1
77da : ff ff ff ff ff ff 00 00  d9
77e2 : ff ff ff ff 00 00 00 00  e1
77ea : ff ff 00 00 00 00 00 00  e9
77f2 : 00 00 00 00 00 00 00 00  f3
77fa : 00 00 00 00 00 00 ff ff  fa
7802 : 00 00 00 00 ff ff ff ff  02
780a : 00 00 ff ff ff ff ff ff  0a
7812 : 78 cc cc 78 30 3c 30 3c  51
781a : cc 78 30 3c 30 3c 00 00  9b
7822 : 30 3c 30 3c 00 00 00 00  04
782a : 30 3c 00 00 00 00 00 00  79
7832 : 00 00 00 00 00 00 00 00  33
783a : 00 00 00 00 00 00 78 cc  b6
7842 : 00 00 00 00 78 cc cc 78  55
784a : 00 00 78 cc cc 78 30 3c  cc
7852 : ff ff ff ff ff ff ff ff  51
785a : 00 00 ff ff ff ff ff ff  5a
7862 : 00 00 00 00 ff ff ff ff  62
786a : 00 00 00 00 00 00 ff ff  6a
7872 : 00 00 00 00 01 00 00 00  73
787a : ff ff 00 00 00 00 00 00  79
7882 : ff ff ff ff 00 00 00 00  81
788a : ff ff ff ff ff ff 00 00  89
7892 : 78 cc cc 78 30 3c 30 3c  d1
789a : 00 00 78 cc cc 78 30 3c  1c
78a2 : 00 00 00 00 78 cc cc 78  b5
78aa : 00 00 00 00 00 00 78 cc  26
78b2 : 00 00 00 00 00 00 00 00  b3
78ba : 30 3c 00 00 00 00 00 00  09
78c2 : 30 3c 30 3c 00 00 00 00  a4
78ca : cc 78 30 3c 30 3c 00 00  4b
78d2 : ff ff ff ff ff ff ff ff  d1
78da : 3f 3f 3f 3f 3f 3f 3f 3f  da
78e2 : 0f 0f 0f 0f 0f 0f 0f 0f  e2
78ea : 03 03 03 03 03 03 03 03  ea
78f2 : 00 00 00 00 00 00 00 00  f3
78fa : c0 c0 c0 c0 c0 c0 c0 c0  f9
7902 : f0 f0 f0 f0 f0 f0 f0 f0  01
790a : fc fc fc fc fc fc fc fc  09
7912 : 78 cc cc 78 30 3c 30 3c  51
791a : 1e 33 33 1e 0c 0f 0c 0f  ea
7922 : 07 0c 0c 07 03 03 03 03  6e
792a : 01 03 03 01 00 00 00 00  8e
7932 : 00 00 00 00 00 00 00 00  33
793a : 00 00 00 00 00 00 00 00  3b
7942 : 80 c0 c0 80 00 c0 00 c0  ea
794a : e0 30 30 e0 c0 f0 c0 f0  e3
7952 : ff ff ff ff ff ff ff ff  51
795a : fc fc fc fc fc fc fc fc  59
7962 : f0 f0 f0 f0 f0 f0 f0 f0  61
796a : c0 c0 c0 c0 c0 c0 c0 c0  69
7972 : 00 00 00 00 00 00 00 00  73
797a : 03 03 03 03 03 03 03 03  7a
7982 : 0f 0f 0f 0f 0f 0f 0f 0f  82
798a : 3f 3f 3f 3f 3f 3f 3f 3f  8a
7992 : 78 cc cc 78 30 3c 30 3c  d1
799a : e0 30 30 e0 c0 f0 c0 f0  33
79a2 : 80 c0 c0 80 00 c0 00 c0  4a
79aa : 00 00 00 00 00 00 00 00  ab
79b2 : 00 00 00 00 00 00 00 00  b3
79ba : 01 03 03 01 00 00 00 00  1e
79c2 : 07 0c 0c 07 03 03 03 03  0e
79ca : 1e 33 33 1e 0c 0f 0c 0f  9a
```

Listing »Schlüssel« (Schluß)





Aufgewickelt

Die LIST-Funktion des VC 20 hat bekanntlich eine Reihe von Nachteilen. Das hier vorgestellte Programm ermöglicht Endloslistings vorwärts und rückwärts und soll als Anregung zur eigenen Beschäftigung mit Betriebssystemroutinen dienen.

Wie unendlich langwierig kann es doch sein, wenn man im Listing eine Stelle sucht, von der man nur ungefähr weiß, wo sie liegt! Manche Basic-Erweiterungen bieten die Möglichkeit, das LISTing per Cursor hin- und herzuschieben. Das gleich zu beschreibende Programm leistet ähnliches. Mehr noch! Es läßt sich durch leichte Ergänzung des einbettenden Basic-Programms beliebig komfortabel gestalten. Im einzelnen bewirkt das Programm nach Listing 1 folgendes:

(1) Ansprung per GOSUB 500. Nach Drücken der Return-Taste Rücksprung ins Ausgangsprogramm oder, falls der Ansprung vom Direktmodus aus erfolgte, in den Direktmodus. Solange das in den Kassettenpuffer gelegte Maschinenprogramm nicht durch andere Operationen (LOAD, SAVE, VERIFY) zerstört wurde, kann jeder weitere Ansprung per GOSUB 580 vorgenommen werden, was eine Einlesezeit von 2,5 sec einspart.

(2) Das LISTing läuft mit der von LIST her bekannten Geschwindigkeit über den Bildschirm (Anfang zufällig), und zwar endlos: Nach Erscheinen der letzten Basic-Zeile erscheint wieder die erste und so weiter. (Die CTRL- und Run/Stop-Taste haben die üblichen Funktionen, sind hier aber überflüssig.)

(3) Das LISTing stoppt nach Drücken der * -Taste und nimmt seinen Lauf nach abermaligem Drücken der * -Taste wieder auf. Alle weiteren Operationen (einschließlich Rückkehr per Return-Taste) können nur am gestoppten LISTing vorgenommen werden.

(4) Drücken der Cursor-Down-Taste läßt die jeweils nächste Basic-Zeile erscheinen (mit der üblichen Repeat-Funktion der Cursor-Taste und dem Bildschirm-Scrollen nach oben bei vollem Bildschirm).

(5) Drücken der Cursor-Right-Taste sucht diejenige Basic-Zeile auf, die auf die letzte auf dem Bildschirm befindliche Zeile folgt, addiert zur Zeilennummer den Wert 4 x 256 und läßt, angefangen bei der sich so ergebenden Zeile, nach Löschung des Bildschirms, fünf aufeinanderfolgende Basic-Zeilen erscheinen. (Der Wert 4 in 256 kann im Programm nach Listing 1 in Zeile 630 abgeändert werden.) Das Ganze bewirkt also, grob gesagt, ein schnelles Durchsuchen des Listings in Sprüngen von 1024 (oder einem vom Benutzer abgeänderten Vielfachen von 256). Ist die Zeile mit der um den Wert 1024 erhöhten Nummer gar nicht vorhanden, was den Normalfall darstellt, da ja 1024 für das Listing als willkürlich gewählter Wert erscheint, wird die darauffolgende Zeile als erste der fünf auszugebenden Zeilen genommen. Genauer gesagt, werden nur vier ausgegeben, da die Suchschleife in diesem Fall einen Leerlaufschritt macht.

(6) Drücken der Cursor-Up-Taste (Cursor mit Shift) löscht den Bildschirm und läßt, ausgehend von der jeweils letzten Basic-Zeile auf dem Bildschirm (genauer: von der auf diese folgenden Zeile), fünf aufeinanderfolgende Basic-Zeilen erscheinen. (Die Zahl 5 kann im Programm nach Listing 1 in Zeile 770 abgeändert werden.) Jedes weitere Drücken der Cursor-Up-Taste nimmt die zur ersten auf dem Bildschirm befindlichen Basic-Zeile vorhergehende Zeile und läßt fünf aufeinanderfolgende Zeilen erscheinen (mit der üblichen Repeat-Funktion der Cursor-Up-Taste). Bei Erreichen der absolut ersten Zeile des LISTings wird dieser Vorgang mit der absolut letzten Zeile fortgesetzt und so weiter. (Rückwärtsverschieben des LISTings über die Anfangszeile hinaus).

Alle verwendeten Cursor-Tasten behalten ihre Repeat-Funktion bei. Das Rückwärtslisten ist dreimal langsamer als das Vorwärtslisten über die * - oder Cursor-Down-Taste und nur zum Einpendeln gedacht: Überblick mit *, Grobabstimmung mit Cursor right, Feinabstimmung mit Cursor down, Korrektur mit Cursor up, endgültige Feinabstimmung mit Cursor down.

Will man das LISTing mit einer vorgegebenen Zeilennummer Y*256+X beginnen lassen, was wegen der oben erwähnten schnellen Durchsuchungsmöglichkeit in Schritten von 1024 an sich nicht nötig ist, so ersetze man den Aufruf GOSUB 500, beziehungsweise GOSUB 580, durch POKE 780, Y:POKE 781, X:GOSUB 500, beziehungsweise POKE 780, Y: POKE 781, X: GOSUB 580.

Für das Maschinenprogramm wird der Trick verwendet, geeignete Teile des Betriebssystems in den Kassettenpuffer zu kopieren und dort durch einige wenige POKE-Anweisungen abzuändern und miteinander zu verbinden. Es werden keine DATA-Zeilen benötigt, wodurch die bekannten Schwierigkeiten mit dem im VC 20-Basic nicht vorhandenen »RESTOREn« umgangen werden.

Neben der eben beschriebenen langen Form des Programms zum Endloslisten nach Listing 1 schlagen wir in Listing 2 noch eine kurze Form vor. Für diese gelten die oben beschriebenen Punkte 1 bis 5 (* -Taste: Lauf, *-Taste: Stillstand, Cursor-Down-Taste: Zeile für Zeile — endlos, Cursor-Right-Taste: Schrittweite 1024, Return-Taste: Rückkehr). Sowohl das Maschinenprogramm als auch das einbettende Basic-Programm sind wesentlich kürzer. Der Hauptaufand wurde im Programm nach Listing 1 für die Organisation des Rückwärtslaufes benötigt. (Fred Behringer/ev)

```
500 REM  ENLOSLISTING                              <031>
501 REM (LANGE FASSUNG)                            <084>
502 REM                                            <134>
503 REM                                            <135>
510 POKE 828,134:POKE 829,20:POKE 830,133          .
   :POKE 831,21                                    <041>
520 FOR I=0 TO 35:POKE 832+I,PEEK(50707+I):NEXT
                          <020>
530 FOR I=0 TO 9:POKE 868+I,PEEK(50952+I):NEXT
                          <250>
540 FOR I=0 TO 8:POKE 879+I,PEEK(50743+I):NEXT
                          <003>
550 FOR I=0 TO 120:POKE 888+I,PEEK(50889+I):NEXT
                          <115>
560 POKE 845,42:POKE 853,34:POKE 865,22
   :POKE 867,20:POKE 875,253:POKE 877,254          <129>
570 POKE 878,136:POKE 887,204:POKE 949,19
   :POKE 963,96                                    <149>
580 GOSUB 780:GET X$:IF X$<>"*"THEN 580            <032>
590 GET X$:IF X$=""THEN 590                        <233>
600 IF X$=CHR$(17)THEN GOSUB 780:GOTO 590          <199>
610 IF X$="*"THEN 580                              <207>
620 IF X$<>CHR$(29)THEN 640                        <157>
630 POKE 780,PEEK(780)+4:GOSUB 770                 <207>
640 IF X$=CHR$(13)THEN 800                         <246>
650 IF X$<>CHR$(145)THEN 590                       <238>
660 X=PEEK(253):Y=PEEK(254)                        <064>
```

Listing 1. Programm für Endloslisting, »lange Form« (vorwärts und rückwärts). Ansprung per GOSUB 500. Einlesezeit für Maschinenprogramm in den Kassettenpuffer 2,5 sec. Jeder weitere Ansprung per GOSUB 580.

```
670 IF X+256*Y<=PEEK(781)+256*PEEK(780)THEN 730
                                           <205>
680 POKE 780,0:POKE 781,0:GOSUB 770        <110>
690 GET X$:IF X$=""THEN 690                <078>
700 IF X$<>CHR$(145)THEN 600               <024>
710 POKE 781,255:POKE 780,249              <002>
720 FOR I=0 TO 3:SYS 828:X=PEEK(253):Y=PEEK(254)
    :POKE 781,X:POKE 780,Y:NEXT            <089>
730 GOSUB 770                              <007>
740 GET X$:IF X$=""THEN 740                <124>
750 IF X$<>CHR$(145)THEN 600               <074>
760 POKE 781,X:POKE 780,Y:GOSUB 780:GOTO 660
                                           <111>
770 PRINT CHR$(147);:FOR I=1 TO 5:GOSUB 780:NEXT
    :RETURN                                <055>
780 SYS 828:IF PEEK(782)=1 THEN POKE 781,0
    :POKE 780,0:GOTO 800                   <012>
790 PRINT CHR$(145);                       <159>
800 RETURN                                 <145>
```

Listing 1. Schluß

Zeile	
510-570	Aufbau des Maschinenprogramms im Kassettenpuffer durch Aneinanderreihung von Teilen des Betriebssystems und leichte Ergänzungen und Abänderungen.
510	Übergabe der X/A-Register (dort steht die Nummer der zu listenden Basic-Zeile) an $0014/$0015 (20/21).
520	$C613-$C636 (50707-50742). Startadresse einer Basic-Programmzeile berechnen. Zeilennummer in $0014/$0015 (20/21).
530	$X708-$C711 (50952-50961). Festhalten der jeweils erreichten Zeilennummer in der Suchschleife und Ablegen nach $00FD/$00FE (6253/254. Nach Erreichen der Adresse der gesuchten Zeilennummer aus $0014/$0015 (20/21) steht in $OOFD/$00FE (253/254) die Nummer der vorhergehenden Zeile. Ist die Nummer in $0014/$0015 (20/21) nicht größer als die erste Zeile des zu listenden Programms, dann wird $C708-$C711 nicht wirksam und $00FD/$00FE (253/254) behält seinen ursprünglichen Wert bei.
540	$C637-$C63F (50743-50751). Weiter in der Suchschleife zur Berechnung der Adresse derjenigen Zeile, deren Nummer in $0014/$0015 (20/21) steht. Die Adresse wird an $005F$/0060 (95/96) ausgegeben. Existiert diese Zeile nicht, dann erscheint die Adresse der nächstfolgenden Zeile. In den X5A-Registern steht jeweils die Nummer (Low-Byte, High-Byte) derjenigen Zeile, welche auf die Zeile folgt, deren Adresse in $005F/$0060 (95/96) gelegt wurde. Ist die in $0014/$0015 (20/21) eingegebene Zeilennummer nicht kleiner als die Nummer der größten Zeile des zu listenden Programms, dann werden X/A mit (für unsere Zwecke) unbrauchbaren Werten belegt; jedoch erhält dann (und nur dann) das Y-Register den Wert 1 (siehe unten Zeile 780). Das Basic-Rahmenprogramm erhält die jeweiligen A-, X-, Y-erte über $030C-$030E (780-782).
550	$C6C9-$C741 (50889-51009). Eigentliches Programm zum Auflisten der gewünschten Zeile. Normalerweise Zeilenvorschub, der in 790 (siehe unten) korrigiert wird. Bei Erreichen der letzten Zeile Austritt ohne Zeilenvorschub, was in Zeile 780 (siehe unten) berücksichtigt wird.
560	Abänderungen von Sprungadressen und Korrektur so, daß die Nummer der jeweils vorhergehenden Zeile in $00FD/$00FE (253/254) abgelegt wird (siehe oben unter Zeile 530).
570	Ergänzung um einen DEY-Befehl $0088 (136). Abänderung zweier Sprungadressen. Sorge dafür, daß nach LISTen der betreffenden Zeile nicht zum Basic-Warmstart gesprungen wird, sondern an die SYS-Rücksprungadresse des aufrufenden Basic-Programms. Das wird durch Überschreiben des Inhalts von $C714 (50964) (natürlich in der Kassettenpuffer-Kopie) mit dem RTS-Befehl $0060 (96) erreicht.
580	Listen, solange nicht *.
590-600	Sonst: Wenn {Cursor down}, dann eine Zeile listen.
610	Wenn wieder *, dann listen, solange nicht abermals *.
620-630	Wenn {Cursor right·, dann Sprung um 1024.
640	Wenn {Return}, dann zurück.
650	Wenn nicht {Cursor up}, dann Neubeginn der Eingabeschleife.
660	Nummer der vorhergehenden Zeile in X/Y.
670-680	Wenn größer, dann Korrektur für Null und fünf Zeilen listen.
690	Neubeginn der Eingabeschleife.
700-720	Wenn nicht {Cursor up}, dann Korrekturen für Nulldurchgang und fünf Zeilen listen.
740	Abermaliger Beginn der Eingangsschleife.
750-760	Wenn wiederum {Cursor up}, dann Vorgänger der Zeile, deren Nummer in X/Y liegt (Anfangszeile des gerade ausgegebenen Fünferblocks) als Anfangszeile des neuen Fünferblocks nehmen und fünf Zeilen ausgeben.
770	Ausgabe von fünf aufeinanderfolgenden Zeilen.
780-800	Ausgabe einer Zeile
790	und Korrektur für Programmende.

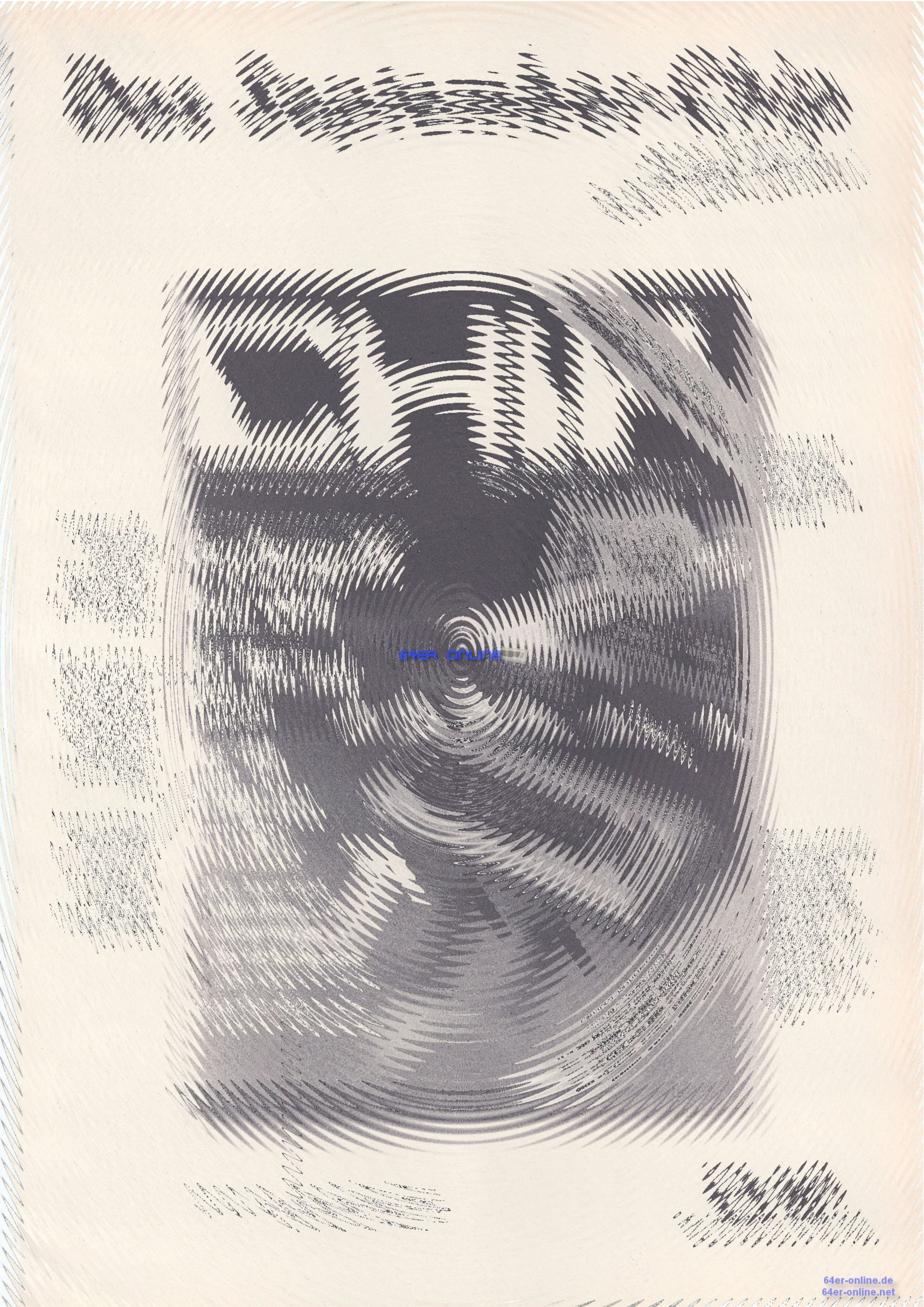
Beschreibung des Programms zum Endloslisten nach Listing 1 (»lange Form«, mit Rückwärtslisten).


```
900 REM  ENDLOSLISTING                     <245>
901 REM (KURZE FASSUNG)                    <016>
902 REM                                    <024>
903 REM                                    <025>
910 POKE 828,134:POKE 829,20:POKE 830,133
    :POKE 831,21                           <187>
920 FOR I=0 TO 45:POKE 832+I,PEEK(50707+I):NEXT
                                           <166>
930 FOR I=0 TO 120:POKE 878+I,PEEK(50889+I):NEXT
                                           <239>
940 POKE 939,19:POKE 953,96                <142>
950 GOSUB 1020:GET X$:IF X$<>"*"THEN 950   <184>
960 GET X$:IF X$=""THEN 960                <093>
970 IF X$=CHR$(17)THEN GOSUB 1020:GOTO 960 <095>
980 IF X$=CHR$(13)THEN 1040                <120>
990 IF X$="*"THEN 950                      <077>
1000 IF X$<>CHR$(29)THEN 960               <031>
1010 POKE 780,PEEK(780)+4:PRINT CHR$(147);
    :FOR I=1 TO 5:GOSUB 1020:NEXT:GOTO 960 <017>
1020 SYS 828:IF PEEK(782)=1 THEN POKE 781,0
    :POKE 780,0:GOTO 1040                  <041>
1030 PRINT CHR$(145);                      <144>
1040 RETURN                                <162>
```

Listing 2. Programm für Endloslisting, »kurze Form« (nicht rückwärts). Ansprung per GOSUB900. Einlesezeit wie in Listing 1. Jeder weitere Ansprung per GOSUB950.

Das Programm Listing 2 ist eine verkürzte Form des Programms nach Listing 1. In Zeile 920 von Listing 2 werden die Zeilen 520 bis 540 von Listing 1 zusammengefaßt. Die Einschubzeile 530 in Listing 1, die die Nummer der jeweils vorhergehenden Zeile zwischenspeichert, wird in Listing 2 nicht benötigt. Die Zeilen 650 bis 760 in Listing 1 organisieren das Rückwärtslisten und werden im Programm nach Listing 2 nicht benötigt.

Beschreibung des Programms zum Endloslisten nach Listing 2 (kurze Form, ohne Rückwärtslisten).



Verbotene Variablen

Hatten Sie schon Probleme mit dem SYNTAX ERROR in anscheinend korrekten Basic-Zeilen? Wahrscheinlich lag es an verbotenen Variablen, die hier näher erklärt werden. Außerdem zeigen wir Ihnen, wie Sie Basic-Programme strukturieren können.

Wer jemals versucht hat, fremde Basic-Programme zu verstehen und dem Gedankengang des Autors zu folgen, der hat sich bestimmt des öfteren darüber geärgert, daß Basic leider allzuoft in einem Spaghetticode ausartet. Sogar der Autor eines Programms findet sich, wenn er nicht 'zig REM-Zeilen eingearbeitet hat, nach einigen Monaten nicht mehr mit dem eigenen Programm zurecht. Das sind dann die Momente, wo man in Richtung Pascal oder all den anderen strukturierten Programmiersprachen schielt.

Der vorliegende Artikel wendet sich an alle, die nach einer Möglichkeit suchen, mit dem vorhandenen (zugegebenermaßen mageren) Basic des C 64 trotzdem ihre Listings etwas übersichtlicher zu gestalten und ihnen eine Struktur zu verleihen.

Übersichtliche Listings

Einige Basic-Programmierer benutzen zur Strukturierung ihrer Programme entweder REMs oder einfach nur den Doppelpunkt. Damit können einzelne Programmteile beim Listing optisch voneinander getrennt werden. Eine bessere Lesbarkeit wird erzielt. Eine andere Möglichkeit zur Strukturierung bieten die Grafikzeichen des C 64.

Nun, werden Sie sich fragen, was denn so besonderes daran sei, Grafikzeichen in Programmen zu verwenden. Man sieht sie doch alle Tage in Listings. Der Witz dieser Lösung zur optischen Aufbereitung von Listings ist, daß die verwendeten Grafikzeichen unsichtbar bleiben, aber die Lesbarkeit der Programme sichtbar unterstützen.

Um diesen scheinbaren Widerspruch aufzulösen, ist es notwendig, sich ein wenig mit der Art und Weise zu beschäftigen, wie der Basic-Interpreter Programmzeilen im Basic-Speicher ab $0801 (dezimal 2049) ablegt.

Schauen Sie sich dazu einmal das kleine Beispielprogramm an.

```
10  FOR I=0 TO 20
20  PRINT"TEST"
30  NEXT
```

Mit einem Monitor erhalten Sie den folgenden Speicherauszug:

```
0800   00   10   08   0A   00   81   20   49
0808   B2   31   20   A4   20   32   30   00
0810   1C   08   14   00   99   22   54   45
0818   53   54   22   00   22   08   1E   00
0820   82   00   00   00
```

Da das Beispielprogramm klein ist, wird noch keine Strukturierung sinnvoll. Stellen Sie sich aber vor, es wäre ein Programm mit 50 und mehr Zeilen, mit einer Menge von GOTOs, GOSUBs und FOR-NEXT-Schleifen. In diesem Fall wird ein herkömmliches Listing von Zeile zu Zeile unübersichtlicher.

Eine einfache, aber recht wirkungsvolle Möglichkeit, ein Listing »durchsichtig« zu machen, ist das Einrücken der Programmzeilen innerhalb einer Schleife. Das Einrücken geschieht meist durch die Voreinstellung eines oder mehrerer Doppelpunkte am Anfang einer Zeile in Verbindung mit einer Reihe von Leerzeichen.

Es geht aber auch anders. Geben Sie dazu die Zeile 20 neu ein:

20 {SHIFT-J} {4*Leertaste} PRINT"TEST"

Wenn Sie nun das kleine Programm listen, ist etwas Erstaunliches passiert:

```
10  FOR I=1 TO 20
20       PRINT"TEST"
30  NEXT
```

Das bei der Eingabe sichtbare Grafikzeichen ist unsichtbar geworden, der PRINT-Befehl aber ist wie bei der Eingabe um vier Leerzeichen (plus ein erzwungenes Leerzeichen nach der Zeilennummer) nach rechts gerückt worden. Wenn Sie das gleiche durch Eingabe von vier Leerschritten unmittelbar nach der Zeilennummer versucht hätten, so wäre der Versuch fehlgeschlagen.

Der Interpreter ignoriert nämlich alle Leerzeichen, die der Programmierer nach der Zeilennummer eingibt; vielmehr wirft er diese alle weg, bis er auf die erste Basic-Aweisung oder einem Doppelpunkt in der betreffenden Zeile trifft. Andererseits besteht er auf genau einem Leerzeichen unmittelbar nach der Zeilennummer, das er eventuell selbst einfügt.

Aber was ist durch das Grafikzeichen im Speicher passiert? Einen Speicherauszug liefert das folgende Bild:

```
0800   00   10   08   0A   00   81   20   49
0808   B2   31   20   A4   20   32   30   00
0810   20   08   14   00   20   20   20   20
0818   99   22   54   45   53   54   22   00
0820   26   08   1E   00   82   00   00   00
```

Die interessantesten Speicherstellen beginnen ab $0810, wo sich gegenüber unserem vorangehenden Beispiel etwas geändert hat. Wir sehen in $0810 und $0811 den Zeiger (hier 20 08 = $0820) auf den Anfang der nächsten Basic-Zeile. Die folgenden Hex-Zeichen 14 00 ergeben in Low-/High-Byte-Darstellung die Zeilennummer 20. Dann folgen die vier eingefügten Leerzeichen ($20) und erst danach das Token 99 (interne Abkürzung des Interpreters) für den PRINT-Befehl. Die folgenden Zeichen sind dann, jeweils um vier Speicherstellen verschoben, mit den Zeichen aus dem vorangehenden Beispiel identisch.

Wenn Sie nun spaßeshalber den Cursor nochmals auf die Zeile 20 des Programmes fahren und die RETURN-Taste drücken, schaut die Zeile 20 nach erneutem Listen wieder so aus, wie ohne Benutzung des »Grafikzeichen-Tricks«. Das geschieht, weil der C 64 einen bildschirmorientierten Editor besitzt, der in letzterem Fall nicht unterscheidet, ob die Zeile 20 neu eingetippt oder »vom Bildschirm übernommen« wird.

Mit dem Eingeben eines Grafikzeichens unmittelbar nach der Zeilennummer übertölpeln Sie den Interpreter, der diese Art von Zeichen an dieser Stelle nicht erwartet hätte. Vor lauter Staunen »vergißt« er dann die eingegebenen Leerzeichen zu ignorieren, wie es sonst seine Art ist.

Verwendung von »verbotenen« Variablen

Hand aufs Herz! Ist es Ihnen nicht auch schon einmal passiert, daß Sie ein Programm geschrieben haben und der Lesbarkeit halber leicht merkbare und selbsterklärende Variablennamen verwendeten. Beim ersten Testlauf stieg der C 64 dann mit der Fehlermeldung SYNTAX ERROR aus und Sie in die Fehlersuche ein.

Meist dauert es dann ein wenig, bis man den Fehler erkennt und beispielsweise einen Variablennamen wie SCHIFF als den Übeltäter entlarvt. SCHIFF ist ein verbotener Name für eine Variable, da der Name eine Basic-Anweisung (IF) enthält. Stößt der Interpreter auf einen solchen Namen, so ist er irritiert, meldet einen Fehler und beendet einfach seine Arbeit.

Schauen Sie sich anhand eines kleinen Beispiels einmal die Arbeitsweise des Interpreters an.

```
10  SCHIFF$="TITANIC"
20  PRINT SCHIFF$
```

Wenn Sie das Programm starten, so geht das Programm ebenso unter wie das besagte Schiff. Ein Monitorlisting schafft Klarheit:

```
0800  00  16  08  0A  00  53  43  48
0808  8B  46  24  B2  22  54  49  54
0810  41  4E  49  43  22  00  23  08
0818  14  00  99  20  53  43  48  8B
0820  46  24  00  00  00
```


Genauer die Speicherzelle $0808 (beziehungsweise $081F). Der C 64 übernimmt eine Programmzeile erst in seinen Programmspeicher, nachdem man die RETURN-Taste gedrückt hat. Dann wird vom Betriebssystem die eingegebene Zeile Zeichen für Zeichen vom Bildschirm übernommen. Der Interpreter hat also beim Eingeben des oben angeführten Programms die Zeichenfolge SCHIFF$ ASCII-Zeichen für Zeichen in die Zellen ab $0805 eingeschrieben, bis er auf die für ihn bekannte Zeichenfolge IF traf. Folgerichtig schrieb er für IF das Token $8B ein und übernahm den Rest der Zeile wie gewohnt. Beim Abarbeiten des Programms versucht der Interpreter, die im Programmspeicher vorgefundenen Zeichen wieder seriell zu lesen und die einzelnen Befehle abzuarbeiten.

Dies mißlingt aber bereits in der ersten Zeile, weil er die Folge (ASCII)SCH(BASIC)IF(ASCII)F$ nicht zu interpretieren vermag. Weil der Interpeter in diesen Fällen so unnachgiebig ist, sind also alle Variablennamen verboten, die Basic-Wörter enthalten (zum Beispiel OR oder TAN).

Beispiele für verbotene Namen sind also: **TAN**NE, **LAND**$, **OR**T$, **STA**RT, um nur einige zu nennen. Wie man an der kleinen Aufzählung erkennt, muß man ziemlich aufpassen, um nicht wieder vom »SYNTAX-TERROR« erwischt zu werden.

Die Lesbarkeit von Programmen leidet schon ein wenig unter dieser Einschränkung, da man sehr oft auf unverständlichere und weniger selbsterklärende Variablennamen ausweichen muß. Mit Hilfe eines kleinen Tricks läßt sich der Interpreter aber so »über's Ohr hauen«, daß er verbotene Variablennamen akzeptiert und dem Programmierer quasi freie Auswahl für seine Variablennamen gibt.

Wer aufmerksam den ersten Teil dieses Artikels gelesen hat, kann sich vielleicht schon denken, wie dieser Trick funktioniert. Richtig, es werden wieder die Grafikzeichen an den entsprechenden Stellen eingesetzt, um den Interpreter von unseren Absichten zu überzeugen.

Geben Sie also ein:

```
10  SCHI{SHIFT-J}F$="TITANIC"
20  PRINT SCHI{SHIFT-J}FF$
```

Ein RUN bestätigt, daß unser Schiff soeben auf den Namen TITANIC getauft wurde und die SYNTAX-ERROR-Meldung des C 64 ausbleibt. Der Monitor zeigt, was mit dem Programm, das sich beim Listen äußerlich in nichts vom vorstehenden Beispiel unterscheidet, geschehen ist.

```
0800  00  17  08  0A  00  53  43  48
0808  49  46  46  24  B2  22  54  49
0810  54  41  4E  49  43  22  00  25
0818  08  14  00  99  20  53  43  48
0820  49  46  46  24  00  00  00
```

Ein Blick auf die Speicherstellen ab $0808 und $0820 zeigt, daß der Interpreter sich überlisten ließ und SCHIFF$ als sieben ASCII-Zeichen gespeichert hat. So kann er das Programm auch ausführen, weil er nicht mehr durch das Basic-Token IF verwirrt wird.

Das Drücken der RETURN-Taste bewirkt, daß der Interpreter jedes einzelne Zeichen der aktuellen Programmzeile übernimmt und auf Zeichenfolgen achtet, die ihm als Basic-Anweisungen oder -Funktionen bekannt sind. Letztere würde er als Token abspeichern. Nachdem er die Zeichenfolge SCH gelesen hat, hat er bereits eine Menge Arbeit hinter sich gebracht.

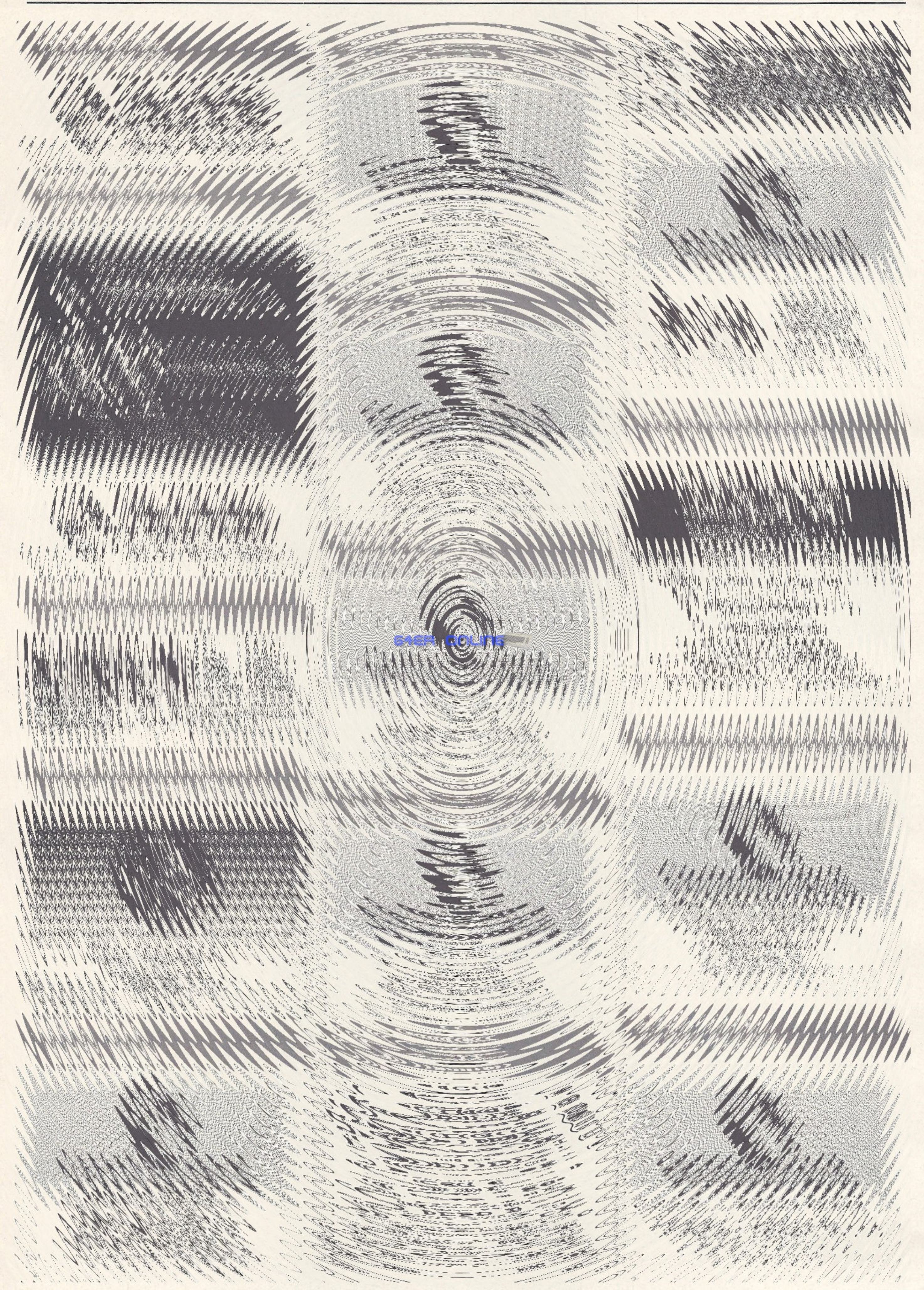
Die Folge SC kennt er nicht, wohl aber die Folge CH. Als nächste Zeichen könnten R und $ dann CHR$ ergeben, was als Token $C7 abgespeichert würde. Aber da er ein I als nächstes Zeichen liest, sagt er sich, vergessen wir die vorangehende Zeichenfolge und legen diese als ASCII-Code im Speicher ab. Bei I allerdings hat der Interpreter wieder seine Ohren gespitzt, denn dies könnte der Anfang von IF (Token $8B) oder auch INPUT (Token $85), INPUT# (Token $84) oder INT (Token $B5) sein.

Der Trick besteht darin, an geeigneter Stelle innerhalb des verbotenen Variablennamens ein Grafikzeichen einzugeben, um die Zeichenfolge so aufzutrennen, daß der Interpreter kein Basic-Wort (Token) erkennen kann.

Bei der Benutzung dieses Tricks ist streng zu beachten, daß jeder Variablenname genau gleich eingegeben wird, damit die Abspeicherung überall gleich ist, wenn ein- und dieselbe Variable mehrmals in einem Programm benutzt wird. Natürlich gilt trotz Benutzung des Tricks mit dem Grafikzeichen weiterhin, daß der C 64 nur die ersten beiden Zeichen eines Variablennamens auswertet. Deshalb ist die Benutzung von TONNE und TORTE im gleichen Programm trotz Eingabe mit unserem Trick nicht eindeutig und kann zu Problemen führen. Weiterhin ist bei zeitkritischen Programmen die Verwendung einstelliger Variablennamen günstiger. Wie weit allerdings die Verwendung von verbotenen Variablen sinnvoll ist, bleibt Ihnen überlassen. War nun das »GOTO« ein Befehl oder eine Variable? Eine Frage, die sich häufiger stellen wird, beim sorglosen Umgang mit verbotenen Variablen. Denkbar ist auch, mit den »verbotenen Variablen« ein Listing nahezu unlesbar zu gestalten (IF GOTO=1 THEN GOTO THEN, etc.), obwohl es optisch wunderbar strukturiert aussieht. Zum Schluß noch eine Bitte der Redaktion. Schicken Sie uns keine derart manipulierten Listings. Es kennt schließlich nicht jeder den Trick mit den Grafikzeichen.

(Dipl.-Ing. Raimund Trierscheid/hm)







Disk-Designer

Dieses Programm für den C 64/VC 20 macht das Inhaltsverzeichnis Ihrer Disketten übersichtlicher und vereinfacht das Laden von Programmen.

Das Programm erklärt sich nach dem Starten von selbst. Es soll aber trotzdem noch kurz auf die einzelnen Funktionen eingegangen werden:

(1) Disketteninhalt wird alphabetisch sortiert.

(2) Files auf der Diskette lassen sich mit einem Löschschutz versehen. Das heißt, sie können nicht mehr versehentlich mit SCRATCH gelöscht werden (beim Formatieren gehen sie allerdings trotzdem verloren).

(3) Dem Programmnamen kann eine Ladehilfe angehängt werden. Das heißt, der Name wird auf zwölf Zeichen verkürzt und dem zweiten Anführungszeichen (") wird die Zeichenfolge »,8:« angehängt. Dies hat den Vorteil, daß nach Laden und Listen des Directorys nur noch vor dem betreffenden Programm »LOAD« und »RETURN« einzugeben ist, um es zu laden.

(4) Alle »SPACEs« in den Programmnamen können mit jedem beliebigen Zeichen aufgefüllt werden. Dadurch ergibt sich ein übersichtlicheres Directory, da alle File-Namen gleich lang dargestellt werden. Beispiel:

Vorher:	Nachher:
64'ER	64'ER . . . .
PROGRAMM	PROGRAMM . . . .
DISK DESIGNER	DISK.DESIGNER . . . .

Mit den Funktionstasten können im Menü die gewünschten Funktionen ein- und ausgeschaltet und die Abarbeitung gestartet werden. Um das Programm auf dem VC 20 lauffähig zu machen, geben Sie bitte die Zeilen aus Listing 2 zusätzlich ein.

(Christian Quirin Spitzner/ev)

Programmbeschreibung

100— 400	: Erklärung
410— 690	: Menü & Auswahl
700— 790	: Dimensionierung von Variablen
800—1190	: Directory lesen
1200—1480	: Files bearbeiten
1490—1750	: Directory sortieren
1760—1970	: Directory schreiben
1980—2070	: Neustart
2080—2230	: Unterprogramm: RAHMEN
2240—2340	: Unterprogramm: Wartet auf »RETURN«
2350—2470	: Unterprogramm: Disk-Errors

```
10 REM***************                        <154>
20 REM*               *                      <069>
30 REM* CHRISTIAN     *                      <117>
40 REM* QUIRIN        *                      <126>
50 REM* SPITZNER      *                      <187>
60 REM* GRUBERSTR.53*                        <209>
70 REM* 8011 POING    *                      <128>
80 REM*               *                      <129>
90 REM***************                        <234>
100 POKE 53280,246                           <027>
110 PRINT"{BLACK}"                           <050>
120 GOSUB 2150                               <004>
130 PRINT"{2RIGHT,DOWN}DIESES PROGRAMM KAN
    N{2SPACE}DAS DIRECTORY"                  <080>
140 PRINT"{2RIGHT,DOWN}IHRER DISKETTE JE{2
    SPACE}NACH{2SPACE}WUNSCH UND"            <238>
150 PRINT"{2RIGHT,DOWN}GESCHMACK NACH VIER
    {2SPACE}KRITERIEN VER-"                  <057>
160 PRINT"{2RIGHT,DOWN}SCHOENERN :           <051>
170 PRINT"{2RIGHT,DOWN}1) DISK{2SPACE}WIRD
    ALPHABETISCH SORTIERT"                   <158>
180 PRINT"{2RIGHT,DOWN}2) FILE- NAMEN{2SPA
    CE}KOENNEN MIT BELIEB-"                  <239>
190 PRINT"{2RIGHT,DOWN,3SPACE}IGEN ZEICHEN
    AUFGEFUELLT WERDEN:"                     <067>
200 PRINT"{2RIGHT,DOWN,3SPACE}Z.B. AUS{2SP
    ACE,RVSON}DISK-DESIGNER{RVOFF}"          <221>
210 PRINT"{2RIGHT,DOWN,8SPACE}WIRD{SPACE,R
    VSON}DISK-DESIGNER...{RVOFF}"            <121>
220 GOSUB 2310                               <008>
230 GOSUB 2150                               <114>
240 PRINT"{2RIGHT,DOWN}3) FILES{2SPACE}KOE
    NNEN{2SPACE}MIT LOESCHSCHUTZ"            <250>
250 PRINT"{2RIGHT,DOWN,3SPACE}VERSEHEN WER
    DEN. D.H. DEM{2SPACE}FILE-"              <045>
260 PRINT"{2RIGHT,DOWN,3SPACE}TYP IM DIREC
    TORY WIRD{2SPACE}EIN{2SPACE}'<'-"        <029>
270 PRINT"{2RIGHT,DOWN,3SPACE}ZEICHEN ANGE
    HAENGT :"                                <184>
280 PRINT"{2RIGHT,DOWN,3SPACE}Z.B OHNE LOS
    CHSCHUTZ : PRG{2SPACE}SEQ "              <215>
290 PRINT"{2RIGHT,DOWN,8SPACE}MIT LOSCHSCH
    UTZ : PRG< SEQ<{6DOWN}"                  <159>
300 GOSUB 2310                               <090>
310 GOSUB 2150                               <196>
320 PRINT"{2RIGHT,DOWN}4) FILES{2SPACE}KOE
    NNEN{2SPACE}MIT{2SPACE}EINER LADE-"      <204>
330 PRINT"{2RIGHT,DOWN,3SPACE}HILFE VERSEH
    EN WERDEN. D.H.{2SPACE}DER"              <000>
340 PRINT"{2RIGHT,DOWN,3SPACE}FILENAME WIR
    D AUF 12 ZEICHEN GE-"                    <255>
350 PRINT"{2RIGHT,DOWN,3SPACE}GEKUERZT UND
    DEM 2. '"CHR$(34)"'- ZEICHEN"            <136>
360 PRINT"{2RIGHT,DOWN,3SPACE}DIE ZEICHENF
    OLGE{SPACE,RVSON},8:{RVOFF,SPACE}ANGEH
    AENGT."                                  <017>
370 PRINT"{2RIGHT,DOWN,3SPACE}SO BRAUCHT M
    AN{2SPACE}NUR NOCH{2SPACE}'LOAD'"        <113>
380 PRINT"{2RIGHT,DOWN,3SPACE}VOR DEN PROG
    RAMMNAMEN ZU SCHREI-"                    <132>
390 PRINT"{2RIGHT,DOWN,3SPACE}UND 'RETURN'
    ZU DRUECKEN UM{2SPACE}EIN"               <218>
400 PRINT"{2RIGHT,DOWN,3SPACE}PROGRAMM ZU
    LADEN.                                   <129>
410 GOSUB 2310                               <200>
420 GOSUB 2150                               <050>
430 G$="JA"                                  <177>
440 E$="JA"                                  <179>
450 I$="JA"                                  <205>
460 F$="."                                   <019>
470 J$="{4SPACE,4LEFT}"                      <038>
480 PRINT"{HOME,6DOWN,2RIGHT,5SPACE,RVSON,
    SPACE}F1{SPACE,RVOFF,SPACE}= START       <129>
485 PRINT"{2RIGHT,DOWN,5SPACE,RVSON,SPACE}
    F2{SPACE,RVOFF,SPACE}= ENDE              <204>
490 PRINT"{2RIGHT,DOWN,5SPACE,RVSON,SPACE}
    F3{SPACE,RVOFF,SPACE}= LOESCHSCHUTZ{4S
    PACE}:";J$;G$                            <156>
500 PRINT"{2RIGHT,DOWN,5SPACE,RVSON,SPACE}
    F5{SPACE,RVOFF,SPACE}= LADEHILFE{7SPAC
    E}:";J$;E$                               <151>
510 PRINT"{2RIGHT,DOWN,5SPACE,RVSON,SPACE}
    F7{SPACE,RVOFF,SPACE}= FILE AUFFUELLEN
     :";J$;I$                                <218>
520 PRINT"{2RIGHT,DOWN,5SPACE,RVSON,SPACE}
    F8{SPACE,RVOFF,SPACE}= ZEICHEN{9SPACE}
    :_{UP,LEFT}U*I{DOWN,LEFT}_{DOWN,3LEFT}
    J*K{UP,2LEFT}";F$;"{LEFT}";              <097>
530 GET D$                                   <000>
540 IF D$=""THEN 530                         <255>
550 E=ASC(D$)                                <145>
560 IF E=133 THEN 700                        <152>
570 IF E=134 AND G$="JA" THEN G$="NEIN":GO
    TO 480                                   <081>
580 IF E=134 AND G$="NEIN" THEN G$="JA":GO
    TO 480                                   <215>
590 IF E=135 AND E$="JA" THEN E$="NEIN":GO
    TO 480                                   <221>
600 IF E=135 AND E$="NEIN" THEN E$="JA":GO
    TO 480                                   <087>
610 IF E=136 AND I$="JA" THEN I$="NEIN":GO
    TO 480                                   <130>
620 IF E=136 AND I$="NEIN" THEN I$="JA":GO
    TO 480                                   <020>
625 IF E=137 THEN PRINT"{CLR}" : END         <012>
```

```
630 IF E<>140 THEN 530                       <152>
640 POKE 204,0                               <031>
650 GET F$:IF F$="" THEN 650                 <155>
660 D=ASC(F$)                                <123>
670 IF D<32 OR (D>127 AND D<160)THEN 650     <215>
680 POKE 204,1                               <199>
690 GOTO 480                                 <006>
700 GOSUB 2150                               <076>
710 IF I$="NEIN" THEN F$=CHR$(160)           <002>
720 DIM PROGR$(224),N(29),M(29)              <177>
730 H=0                                      <151>
740 O=0                                      <189>
750 P=1                                      <219>
760 G=0                                      <177>
770 N=18                                     <240>
780 M=1                                      <239>
790 K$=CHR$(160)+",8:"                       <178>
800 :                                        <119>
810 :                                        <024>
820 :                                        <034>
830 REM  ***   DIRECTORY LESEN   ***         <151>
840 :                                        <054>
850 :                                        <064>
860 :                                        <074>
870 PRINT"{2DOWN,RIGHT,7SPACE}DIRECTORY WI
    RD GELESEN"                              <085>
880 PRINT"{14DOWN,RIGHT,9SPACE}U**********
    ******I"                                 <027>
890 PRINT"{RIGHT,9SPACE}-{16SPACE}-"         <011>
900 PRINT"{RIGHT,9SPACE}J****************K
    "                                        <042>
910 OPEN 15,8,15,"I0"                        <021>
920 GOSUB 2420                               <250>
930 OPEN 2,8,2,"#"                           <177>
940 GOSUB 2420                               <014>
950 PRINT#15,"U1";2;0;N;M                    <038>
960 N(O)=N                                   <044>
970 M(O)=M                                   <177>
980 GET#2,EG$                                <057>
990 IF EG$="" THEN EG$=CHR$(0)               <208>
1000 N=ASC(EG$)                              <151>
1010 GET#2,EG$                               <087>
1020 IF EG$="" THEN EG$=CHR$(0)              <238>
1030 M=ASC(EG$)                              <179>
1040 L=2                                     <001>
1050 PRINT#15,"B-P";2;L                      <089>
1060 H=H+1                                   <007>
1070 FOR I=1 TO 30                           <201>
1080 :  GET#2,EG$                            <233>
1090 :  IF EG$="" THEN EG$=CHR$(0)           <013>
1100 :  PROGR$(H) = PROGR$(H) + EG$          <084>
1110 NEXT I                                  <178>
1120 PRINT"{HOME,22DOWN}";TAB(11);MID$(PRO
     GR$(H),4,16)                            <084>
1130 IF ASC(LEFT$(PROGR$(H),1))<>0 THEN 11
     80                                      <238>
1140 PROGR$(H)=""                            <040>
1150 H=H-1                                   <129>
1160 G=G+1                                   <087>
1170 P=0                                     <115>
1180 IF L-226 THEN L=L+32:GOTO 1050          <086>
1190 IF N THEN L=2 : O=O+1 : GOTO 950        <183>
1200 :                                       <160>
1210 :                                       <170>
1220 :                                       <180>
1230 REM ***   FILES BEARBEITEN   ***        <001>
1240 :                                       <200>
1250 :                                       <210>
1260 :                                       <220>
1270 FOR J=1 TO H                            <244>
1280 PRINT"{HOME,8DOWN,2RIGHT,6SPACE,RVSON
     }FILES WERDEN BEARBEITET"               <148>
1290 FOR J=1 TO H                            <010>
1300 :  C$=MID$(PROGR$(J),17,1)              <241>
1310 :  FOR I=4 TO 20                        <247>
1320 :      A$=MID$(PROGR$(J),I,1)           <004>
1330 :      IF A$=CHR$(160) AND E$="JA" THE
     N A$=F$ : GOTO 1370                     <009>
1340 :      IF A$=CHR$(160) AND C$<>"," THE
     N A$=F$ : GOTO 1370                     <104>
1350 :      IF A$=CHR$(32) AND I$="NEIN" TH
     EN A$=CHR$(32) : GOTO 1370              <233>
1360 :      IF A$=CHR$(32) AND I$="JA" THEN
       A$=F$                                 <071>
1370 :      PROGR$(J)=LEFT$(PROGR$(J),I-1)+
     A$+RIGHT$(PROGR$(J),30-I)               <109>
1380 :   NEXT I                              <254>
1390 :   F=ASC(LEFT$(PROGR$(J),1))           <245>
1400 :   IF F<150 AND G$="NEIN"THEN 1460     <223>
1410 :   IF F>150 AND G$="JA"THEN 1460       <253>
1420 :   IF G$="JA"THEN F=F+64:GOTO 1440     <080>
1430 :   IF G$="NEIN"THEN F=F-64             <030>
1440 :   PROGR$(J)=CHR$(F)+RIGHT$(PROGR$(J)
     ,29)                                    <241>
1450 :   IF F >< 130 AND F >< 194 THEN 1470  <115>
1460 :   IF E$="JA"THEN PROGR$(J)=LEFT$(PRO
     GR$(J),15)+K$+RIGHT$(PROGR$(J),11)      <184>
1470 PRINT"{HOME,22DOWN,RVSON}";TAB(11);MI
     D$(PROGR$(J),4,16)                      <079>
1480 NEXT J                                  <046>
1490 :                                       <196>
1500 :                                       <206>
1510 :                                       <216>
1520 REM *** DIRECTORY SORTIEREN ***         <183>
1530 :                                       <236>
1540 :                                       <248>
1550 :                                       <002>
1560 PRINT"{HOME,10DOWN,2RIGHT,6SPACE}FILE
     S WERDEN SORTIERT"                      <064>
1570 FOR J=1 TO H-1                          <049>
1580 :   FOR I=J+1 TO H                      <152>
1590 :      IF MID$(PROGR$(I),4,16)>MID$(PR
     OGR$(J),4,16)THEN 1640                  <046>
1600 :      H$=PROGR$(I)                     <135>
1610 :      PROGR$(I)=PROGR$(J)              <159>
1620 :      PROGR$(J)=H$                     <245>
1630 :      P=0                              <005>
1640 :   NEXT I                              <004>
1650 :   PRINT"{HOME,22DOWN}";TAB(11);MID$(
     PROGR$(J),4,16)                         <107>
1660 NEXT J                                  <228>
1670 IF P THEN 2050                          <216>
1680 H$=""                                   <213>
1690 IF G=0 THEN 1830                        <030>
1700 FOR J=1 TO 30                           <075>
1710 :   H$=H$+CHR$(0)                       <140>
1720 NEXT J                                  <032>
1730 FOR J=1 TO G                            <067>
1740 :   PROGR$(H+J)=H$                      <241>
1750 NEXT J                                  <062>
1760 :                                       <212>
1770 :                                       <222>
1780 :                                       <232>
1790 REM *** DIRECTORY SCHREIBEN ***         <189>
1800 :                                       <254>
1810 :                                       <008>
1820 :                                       <018>
1830 PRINT"{HOME,12DOWN,2RIGHT,6SPACE,RVSO
     N}DIRECTORY WIRD GESCHRIEBEN"           <214>
1840 K=1                                     <019>
1850 FOR J=0 TO O                            <161>
1860 :   PRINT#15,"B-P";2;0                  <160>
1870 :   PRINT#2,CHR$(N(J+1));CHR$(M(J+1))   <127>
1880 :   L=2                                 <033>
1890 :   FOR I=1 TO 8                        <036>
1900 :      PRINT#15,"B-P";2;L               <214>
1910 :      PRINT#2,PROGR$(K);               <046>
1920 :      PRINT"{HOME,22DOWN,RVSON}";TAB(
     11);MID$(PROGR$(K),4,16)                <180>
1930 :      K=K+1                            <176>
1940 :      L=L+32                           <021>
1950 :   NEXT I                              <060>
1960 :   PRINT#15,"U2";2;0;N(J);M(J)         <072>
1970 NEXT J                                  <028>
1980 :                                       <178>
1990 :                                       <188>
2000 :                                       <198>
2010 REM ***       NEUSTART       ***        <097>
2020 :                                       <218>
2030 :                                       <228>
2040 :                                       <238>
2050 CLOSE 2                                 <037>
2060 CLOSE 15                                <122>
2070 RUN 420                                 <026>
2080 :                                       <124>
2090 :                                       <034>
2100 :                                       <044>
2110 REM ***        RAHMEN        ***        <210>
2120 :                                       <064>
2130 :                                       <074>
```

Listing 1. »Disk-Designer«, Version für den C 64

```
2140 :                                         <084>
2150 PRINT"{CLR}U**************************
     *************I"                           <028>
2160 FOR I=1 TO 23                             <150>
2170 :  PRINT"_{37SPACE}_"                     <163>
2180 NEXT I                                    <232>
2190 PRINT"J*******************************
     ********K{HOME}"                          <139>
2200 PRINT"{5RIGHT}U***********************
     *****I"                                   <000>
2210 PRINT"{5RIGHT}_{RVSON,SPACE}* * * DIS
     K-DESIGNER * * *{SPACE,RVOFF}_"           <254>
2220 PRINT"{5RIGHT}J***********************
     *****K"                                   <062>
2230 RETURN                                    <000>
2240 :                                         <184>
2250 :                                         <194>
2260 :                                         <204>
2270 REM *** WARTET [RETURN] AB   ***          <069>
2280 :                                         <224>
2290 :                                         <234>
2300 :                                         <244>
2310 PRINT"{2RIGHT,DOWN}DRUECKE [{RVSON}RE
     TURN{RVOFF}]                              <230>
2320 GET RETUR$                                <186>
2330 IF RETUR$ <> CHR$(13) THEN 2320           <074>
2340 RETURN                                    <112>
2350 :                                         <040>
2360 :                                         <050>
2370 :                                         <060>
2380 REM ***      DISK ERROR       ***         <078>
2390 :                                         <080>
2400 :                                         <090>
2410 :                                         <100>
2420 INPUT#15,A,B$,B,C                         <213>
2430 IF A=0 THEN RETURN                        <041>
2440 GOSUB 2150                                <038>
2450 PRINT"{3DOWN,RIGHT,5SPACE,RVSON}DISK-
     ERROR :"                                  <120>
2460 PRINT"{DOWN,RIGHT,4SPACE}";A;B$;B;C       <073>
2470 GOTO 2050                                 <036>
```

Listing 1. »Disk-Designer«, Version für den C 64 (Schluß)

```
100 POKE 36879,25                              <146>
110 PRINT"{BLUE}"                              <026>
130 PRINT"{RIGHT}DIESES{5SPACE}PROGRAMM"       <027>
135 PRINT"{RIGHT}KANN DAS{2SPACE}DIRECTORY
    "                                          <090>
140 PRINT"{RIGHT}IHRER{2SPACE}DISKETTE{2SP
    ACE}JE"                                    <232>
145 PRINT"{RIGHT}NACH{3SPACE}WUNSCH{3SPACE
    }UND"                                      <007>
150 PRINT"{RIGHT}GESCHMACK NACH VIER"          <062>
155 PRINT"{RIGHT}KRITERIEN{6SPACE}VER-"        <125>
160 PRINT"{RIGHT}SCHOENERN :                   <254>
170 PRINT"{RIGHT}1) DISK{3SPACE}WIRD{2SPAC
    E}AL-"                                     <023>
175 PRINT"{RIGHT}PHABETISCH SORTIERT"          <196>
180 PRINT"{RIGHT}2) FILE-NAMEN KOEN-"          <107>
185 PRINT"{RIGHT}NEN{2SPACE}MIT BELIEBIGEN
    "                                          <152>
190 PRINT"{RIGHT}ZEICHEN AUFGEFUELLT"          <085>
195 PRINT"{RIGHT}WERDEN: Z.B. AUS              <144>
200 PRINT"{RIGHT,2SPACE,RVSON}DISK-DESIGNE
    R{RVOFF}"                                  <080>
205 PRINT"{RIGHT,13SPACE}WIRD                  <190>
210 PRINT"{RIGHT,2SPACE,RVSON}DISK-DESIGNE
    R...{RVOFF,DOWN}"                          <010>
240 PRINT"{2RIGHT,DOWN}3)FILES{4SPACE}KOEN
    NEN"                                       <151>
245 PRINT"{RIGHT,3SPACE}MIT LOESCHSCHUTZ"      <228>
250 PRINT"{RIGHT,3SPACE}VERSEHEN WERDEN."      <144>
255 PRINT"{RIGHT,3SPACE}D.H.{2SPACE}DEM{2S
    PACE}FILE-"                                <102>
260 PRINT"{RIGHT,3SPACE}TYP IM DIRECTORY"      <196>
265 PRINT"{RIGHT,3SPACE}WIRD{3SPACE}EIN{2S
    PACE}'<'-"                                 <151>
270 PRINT"{RIGHT,3SPACE}ZEICHEN{4SPACE}ANG
    E-"                                        <179>
275 PRINT"{RIGHT,3SPACE}HAENGT:                <086>
280 PRINT"{RIGHT,3SPACE}OHNE LOSCHSCHUTZ{D
    OWN}                                       <016>
285 PRINT"{RIGHT,3SPACE}PRG{2SPACE}SEQ{2SP
    ACE}REL{DOWN}                              <125>
290 PRINT"{RIGHT,3SPACE}MIT{2SPACE}LOSCHSC
    HUTZ{DOWN}                                 <178>
295 PRINT"{RIGHT,3SPACE}PRG< SEQ< REL<{DOW
    N}"                                        <157>
320 PRINT"{RIGHT}4)FILES{5SPACE}KOENNEN"       <068>
325 PRINT"{RIGHT,2SPACE}MIT{2SPACE}EINER{2
    SPACE}LADE-"                               <053>
330 PRINT"{RIGHT,2SPACE}HILFE{4SPACE}VERSE
    HEN"                                       <213>
335 PRINT"{RIGHT,2SPACE}WERDEN. D.H.{2SPAC
    E}DER"                                     <207>
340 PRINT"{RIGHT,2SPACE}FILENAME WIRD AUF"     <086>
345 PRINT"{RIGHT,2SPACE}12{2SPACE}ZEICHEN{
    3SPACE}GE-"                                <169>
350 PRINT"{RIGHT,2SPACE}KUERZT{2SPACE}UND{
    3SPACE}DEM"                                <185>
355 PRINT"{RIGHT,2SPACE}2.{2SPACE}'"CHR$(3
    4)"' - ZEICHEN"                            <149>
360 PRINT"{RIGHT,2SPACE}DIE{2SPACE}ZEICHEN
    FOLGE"                                     <131>
365 PRINT"{RIGHT,2SPACE,RVSON},8:{RVOFF,3S
    PACE}ANGEHAENGT."                          <138>
370 PRINT"{RIGHT,2SPACE}SO{2SPACE}BRAUCHT{
    3SPACE}MAN"                                <240>
375 PRINT"{RIGHT,2SPACE}NUR{2SPACE}NOCH{2S
    PACE}'LOAD'"                               <203>
380 PRINT"{RIGHT,2SPACE}VOR DEN PROGRAMM-"     <184>
385 PRINT"{RIGHT,2SPACE}NAMEN{2SPACE}ZU SC
    HREI-"                                     <242>
390 PRINT"{RIGHT,2SPACE}UND{2SPACE}'RETURN
    '{2SPACE}ZU"                               <002>
395 PRINT"{RIGHT,2SPACE}DRUECKEN{2SPACE}UM
    {2SPACE}EIN"                               <173>
400 PRINT"{RIGHT,2SPACE}PROGRAMM ZU LADEN"     <133>
480 PRINT"{HOME,4DOWN,RIGHT,DOWN,RVSON}F1{
    RVOFF,SPACE}=START{SPACE,RVSON}F2{RVOF
    F,SPACE}=ENDE                              <165>
485 :                                          <207>
490 PRINT"{RIGHT,DOWN,RVSON}F3{RVOFF,SPACE
    }=LOESCHSCHUTZ"                            <127>
495 PRINT TAB(9);J$;G$                         <247>
500 PRINT"{RIGHT,DOWN,RVSON}F5{RVOFF,SPACE
    }=LADEHILFE                                <236>
505 PRINT TAB(9);J$;E$                         <249>
510 PRINT"{RIGHT,DOWN,RVSON}F7{RVOFF,SPACE
    }=FILE AUFFUELLEN"                         <105>
515 PRINT TAB(9);J$;I$                         <021>
520 PRINT"{RIGHT,DOWN,RVSON}F8{RVOFF,SPACE
    }=ZEICHEN{DOWN}"                           <231>
525 PRINT TAB(9)"_{UP,LEFT}U*I{DOWN,LEFT}_
    {DOWN,3LEFT}J*K{UP,2LEFT}";F$;"{LEFT}"
    ;                                          <247>
870 PRINT"{2DOWN,RIGHT}DIRECTORY LESEN"        <126>
880 PRINT"{12DOWN,RIGHT}U****************I
    "                                          <230>
890 PRINT"{RIGHT}_{16SPACE}_"                  <011>
900 PRINT"{RIGHT}J****************K"           <042>
1120 PRINT"{HOME,20DOWN,2RIGHT}"MID$(PROGR
     $(H),4,16)                                <125>
1280 PRINT"{HOME,8DOWN,RIGHT,RVSON}FILES B
     EARBEITEN"                                <007>
1470 PRINT"{HOME,20DOWN,2RIGHT,RVSON}"MID$
     (PROGR$(J),4,16)                          <183>
1560 PRINT"{HOME,10DOWN,RIGHT}FILES SORTIE
     REN"                                      <190>
1650 :  PRINT"{HOME,20DOWN,2RIGHT}"MID$(PR
     OGR$(J),4,16)                             <027>
1830 PRINT"{HOME,12DOWN,RIGHT,RVSON}DIRECT
     ORY SCHREIBEN"                            <092>
1920 :      PRINT"{HOME,20DOWN,2RIGHT,RVSON
     }"MID$(PROGR$(K),4,16)                    <224>
2150 PRINT"{CLR}U*******************I"         <071>
2160 FOR I=1 TO 21                             <148>
2170 :  PRINT"_{19SPACE}_"                     <163>
2180 NEXT I                                    <232>
2190 PRINT"J*******************K{HOME}"        <222>
2200 PRINT"{RIGHT}U*****************I"         <220>
2210 PRINT"{RIGHT}_{RVSON}* DISK-DESIGNER
     *{RVOFF}_"                                <119>
2220 PRINT"{RIGHT}J*****************K"         <144>
2310 PRINT"{3RIGHT}DRUECKE [{RVSON}RETURN{
     RVOFF}]                                   <233>
```

Listing 2. Diese Zeilen müssen für den VC 20 in Listing 1 geändert werden.

Blinker

Das Programm ist etwa 130 Byte lang und frei im Speicher des C 64 verschiebbar. Mit dem Befehl

SYS Startadresse, Anzahl der Menüpunkte (maximal 9), Anfangszeile werden Zahlen von 1 bis 9 am linken Bildschirmrand dargestellt. Diese Zahlen blinken so lange, bis eine dieser Zahlen eingegeben wird. Fehleingaben werden ignoriert. Die eingegebene Zahl wird im ASCII-Format in die Speicherstelle 251 geschrieben und kann mit dem Befehl PEEK(251) vom Basic ausgelesen und weiter verarbeitet werden. Bei einem Fehler in dem SYS-Befehl wird eine »0« in die Speicherstelle 251 geschrieben.

Nach dem Start mit RUN muß dem C 64 mitgeteilt werden, ab welcher Adresse das Programm stehen soll.

Was das Programm leistet, zeigt das folgende Demoprogramm. Dazu müssen Sie in Zeile 50 der Variablen BL die Startadresse des Maschinenprogramms zuordnen.

(P.K.Bhattacharya/ah)

```
10 A$=CHR$(18):A$=A$+CHR$(176)                  <080>
20 FOR I=1 TO 38:A$=A$+CHR$(192):NEXT:A$=A
   $+CHR$(174):A$=A$+CHR$(221)                  <191>
30 FOR I=1 TO 38:A$=A$+CHR$(32):NEXT:A$=A$
   +CHR$(221):A$=A$+CHR$(173)                   <127>
40 FOR I=1 TO 38:A$=A$+CHR$(192):NEXT:A$=A
   $+CHR$(189)                                  <148>
50 :BL=828:S$="{HOME,6DOWN}":L$="{4SPACE}"
   :V=53280                                     <023>
60 FOR I=0 TO 3:READ A$(I):NEXT:FOR I=0 TO
    3:READ B$(I):NEXT                           <064>
70 FOR I=0 TO 3:READ T$(I):NEXT:FOR I=0 TO
    3:READ F$(I):NEXT                           <166>
80 FOR I=0 TO 5:READ D$(I):NEXT                 <163>
90 :POKE V,6:PRINT"{CLR}";CHR$(14)CHR$(8)C
   HR$(5)                                       <053>
100 X$=B$(0):GOSUB 500:PRINT S$;:FOR I=0 T
    O 3:PRINT L$A$(I):PRINT:NEXT:SYS BL,4,
    6                                           <172>
110 FB=VAL(CHR$(PEEK(251))):ON FB GOTO 150
    ,200,250,2000                               <071>
120 GOTO 100                                    <048>
150 X$=B$(1):GOSUB 500:PRINT S$;:FOR I=0 T
    O 3:PRINT L$T$(I):PRINT:NEXT:SYS BL,4,
    6                                           <105>
160 FB=VAL(CHR$(PEEK(251))):ON-(FB=4)GOTO
    100:GOTO 400                                <039>
200 X$=B$(2):GOSUB 500:PRINT S$;:FOR I=0 T
    O 3:PRINT L$F$(I):PRINT:NEXT:SYS BL,4,
    6                                           <149>
210 FB=VAL(CHR$(PEEK(251))):ON-(FB=4)GOTO
    100:GOTO 400                                <089>
250 X$=B$(3):GOSUB 500:PRINT S$;:FOR I=0 T
    O 5:PRINT L$D$(I):PRINT:NEXT:SYS BL,6,
    6                                           <015>
260 FB=VAL(CHR$(PEEK(251))):ON-(FB=6)GOTO
    100:GOTO 400                                <143>
400 PRINT"{HOME}":FOR I=0 TO 18:PRINT:NEXT
    :PRINT L$L$"BITTE EINE TASTE DRUCKEN"       <242>
410 POKE 198,0:WAIT 198,255:GOTO 100            <187>
500 REM                                         <052>
510 LN=LEN(X$):PRINT"{CLR}";A$;"{HOME,DOWN
    }";SPC((40-LN)/2);X$CHR$(146);"{2DOWN}
    ":RETURN                                    <247>
1000 DATA "ARBEIT MIT TASTATUR","ARBEIT MI
     T DISKETTE","ARBEIT MIT DRUCKER"           <054>
1010 DATA "ENDE","MENUE","TASTATUR","DISKE
     TTE","DRUCKER"                             <083>
1020 DATA "LESEN","SCHREIBEN","EDITIEREN",
     "MENUE"                                    <175>
1030 DATA "DIRECTORY","LOAD","SAVE","MENUE
     "                                          <124>
1040 DATA "BRIEFKOPF","GRUSSWORT","TITEL",
     "SEITENZAHL","TEXT","MENUE"                <197>
2000 POKE V,14:PRINT CHR$(9):END                <138>
```

© 64'er

Listing 2. Demo-Listing zum Programm Blinker. Bei der Eingabe bitte Seite 54 beachten

```
10 PRINT"{CLR,WHITE}"                           <226>
11 INPUT "STARADRESSE=";BL                      <055>
20 FOR I=BL TO BL+132:READ X:S=S+X:POKE I,
   X:NEXT:IF S<>19188 THEN PRINT"FEHLER":G
   OTO 107                                      <161>
100 DATA 32,253,174,32,158,183,134,253,224
    ,10,176,113,224,1,144,109,32,253            <175>
101 DATA 174,32,158,183,134,254,165,253,10
    ,24,101,254,201,26,176,91,165,253           <001>
102 DATA 9,48,133,252,166,254,160,0,24,32,
    240,255,169,48,133,250,230,250,165          <190>
103 DATA 2,240,4,169,18,133,199,165,250,32
    ,210,255,169,13,32,210,255,165,250          <045>
104 DATA 197,252,240,7,169,13,32,210,255,2
    08,223,162,103,165,198,240,15,32            <217>
105 DATA 180,229,133,251,197,252,240,29,17
    6,4,201,49,176,23,136,208,234,202           <085>
106 DATA 208,231,165,2,240,5,138,133,2,240
    ,175,230,2,208,171,169,0,133,251            <024>
107 DATA 96,0,0,0:POKE V,14:END                 <245>
```

© 64'er

Listing 1. Listing zum Programm Blinker. Bei der Eingabe bitte Seite 54 beachten

Hypra-Load für SX 64

Wie bereits in der ersten Beschreibung von Hypra-Load (Ausgabe 10/84) erwähnt, funktioniert das Programm nicht auf dem Commodore SX 64 trotz angeblicher Vollkompatibilität. Der Grund hierfür liegt in der unterschiedlichen Verwendung der Speicherstelle 1 bei den Computern C 64 und SX 64.

Hier ist erstens die Lage des Basic-Interpreters, des Kernals, der Ein-/Ausgabe-Register und des Zeichengenerators codiert (Genaueres siehe »Memory-Map mit Wandervorschlägen«, Teil 1, Ausgabe 11/84). Zweitens wird jedoch auch die Datasette über diese Speicherstelle gesteuert. An den SX 64 kann nun aber keine Datasette angeschlossen werden. Deswegen dürfen die Bits, die die Datasette betreffen, auch nicht gesetzt werden. Daraus ergibt sich folgendes: Soll das RAM unter dem ROM aktiviert werden, darf in die Speicherstelle 1 keine 53 gePOKEt werden, sondern nur eine 5. Ähnliches gilt für das Einschalten des ROM. Hier muß eine 7 anstelle einer 55 verwendet werden.

Hypra-Load kopiert das Betriebssystem ins RAM, ändert und aktiviert es dann. Soll dies nun auch auf dem SX 64 funktionieren, müssen die Befehle, die die Speicherstelle 1 betreffen, angepaßt werden.

Zur Anpassung laden Sie bitte Hypra-Load und geben Sie folgende sechs POKE-Befehle ein:

POKE 2407,5 : POKE 2422,5 : POKE 3387,5
POKE 2471,7 : POKE 3114,7 : POKE 3240,7

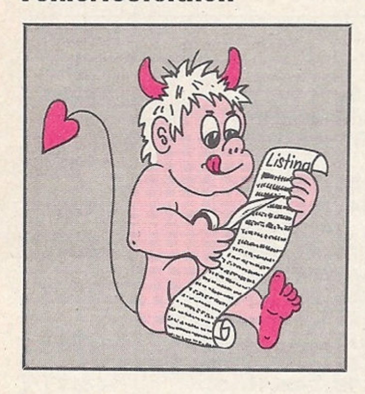
Das veränderte Hypra-Load können Sie nun einfach mit dem SAVE-Befehl abspeichern.

Übrigens müßten alle Versionen von Hypra-Load, die im Betriebssystem-ROM liegen (Hypra-Perfekt; Hypra-Centronics), einwandfrei auf dem SX 64 funktionieren, da hier ja keine Umschaltung auf das RAM erfolgt.

(Andreas Zander/bs)



Fehlerteufelchen



Reset-Taster für alle Fälle, Ausgabe 6/85, Seite 131

In Bild 2 muß es natürlich »Expansion-Port« und nicht »User-Port« heißen.

Grafik auf dem C 64, Ausgabe 5/85, Seite 86

Der Preis dieses Buches beträgt nicht wie angegeben 39 Mark, sondern 29,80 Mark.

PRO.ST mit dem C 64, Ausgabe 6/85, Seite 70

In der Programmbeschreibung wurde der Hinweis vergessen, daß der Programmablauf jederzeit mit »Shift« abgebrochen werden kann.

Radar, SH 3 »Spiele«, Seite 31

Unter »Hinweise zum Abtippen« muß es in der fünften Zeile heißen: »POKE194,32« anstatt »POKE32,196«.

Spiralen mit dem Plotter 1520, Ausgabe 4/85, Seite 155

Die zweite PRINT-Anweisung darf nicht lauten: »PRINT#1,"S" «..., da es gar keinen "S"-Befehl zum Zeichen von Koordinaten gibt. Das Programm funktioniert einwandfrei, wenn man das »S« gegen ein »D« oder ein »J« austauscht.

REM-Killer, Ausgabe 7/85, Seite 75

In den REM-Killer hat sich ein logischer Fehler eingeschlichen. Sollen GOTO- und GOSUB-Befehle direkt hinter einem THEN-Befehl angeglichen werden, sind folgende Änderungen nach dem Laden des REM-Killers nötig:

»POKE 52621,154:POKE52625,150«

Tiny Forth Compiler, Ausgabe 8/85

Selbst das Listing des Monats ist nicht vor dem Fehlerteufelchen sicher: Beim Tiny Forth Compiler funktioniert die Variablen-Definition in der abgedruckten Form leider nicht vorschriftsmäßig, sondern führt im Gegenteil zum Programmabsturz. Die hier wiedergegebene Neufassung der VARIABLE-Routine behebt den Fehler.

```
1080 IF BE$<>"VARIABLE" THEN 1145
1085 GOSUB 2630:AN=AN+1:WO$(AN)=BE$
1090 AD(AN)=VOC:XX=VOC+8
1095 GOSUB 3470:POKE VOC,169
1100 POKE VOC+1,FN H(XX)
1105 POKE VOC+2,162
1110 POKE VOC+3,FN L(XX)
1115 POKE VOC+4,32:POKE VOC+5,42
1120 POKE VOC+6,192:POKE VOC+7,96
1125 POKE VOC+8,FN L(X)
1130 POKE VOC+9,FN H(X)
1135 VOC=VOC+10
1140 GOTO 700
READY.
```

Listing der neuen VARIABLE-Routine

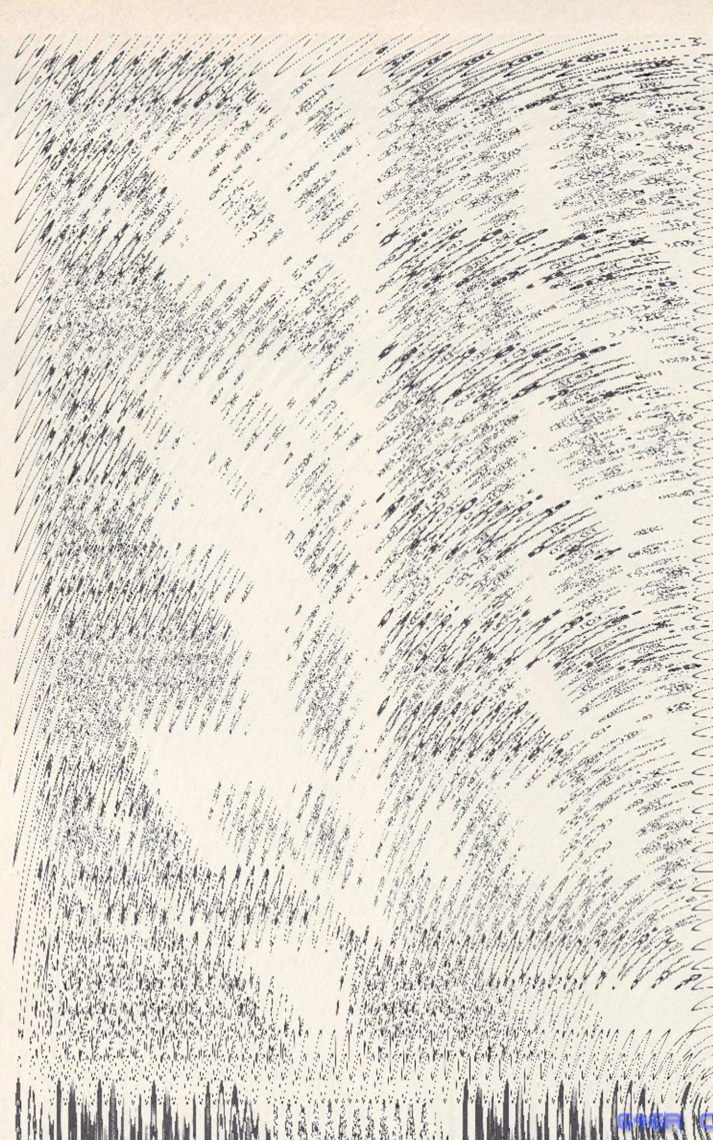
Wollen Sie einen gebrauchten Computer verkaufen oder erwerben? Suchen Sie Zubehör? Haben Sie Software anzubieten oder suchen Sie Programme oder Verbindungen? Der COMPUTER-MARKT von »64er« bietet allen Computerfans die Gelegenheit, für nur 5,— DM eine private Kleinanzeige mit bis zu 5 Zeilen Text in der Rubrik Ihrer Wahl aufzugeben. Und so kommt Ihre private Kleinanzeige in den COMPUTER-MARKT der **Oktober-Ausgabe** (erscheint am 20. September 85): Schicken Sie Ihren Anzeigentext bis zum 28. August 85 (Eingangsdatum beim Verlag) an »64er«. Später eingehende Aufträge werden in der **November-Ausgabe** (erscheint am 18. 10. 85) veröffentlicht.

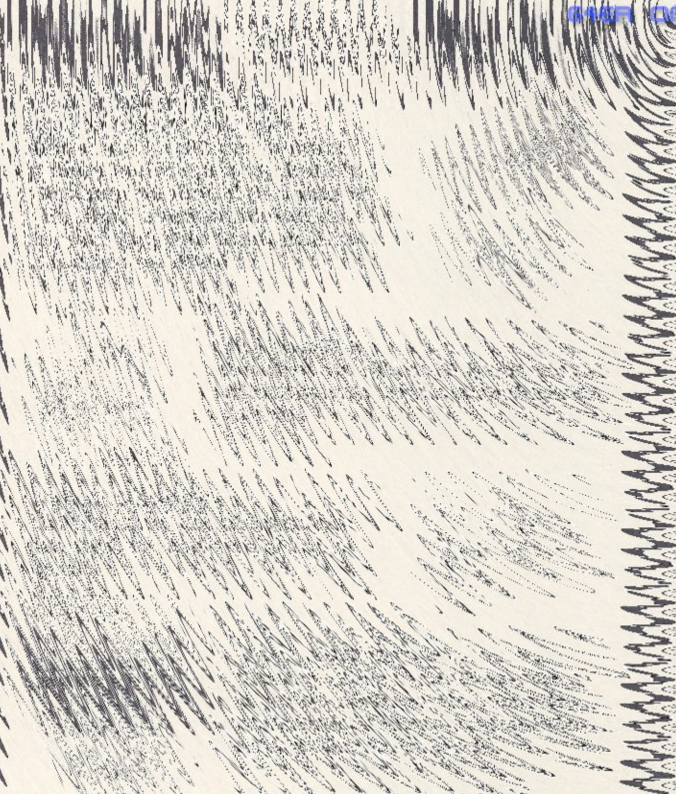
Am besten verwenden Sie dazu die vorbereitete Auftragskarte am Anfang des Heftes. Bitte beachten Sie: Ihr Anzeigentext darf maximal 5 Zeilen mit je 32 Buchstaben betragen. Überweisen Sie den Anzeigenpreis von DM 5,— auf das Postscheckkonto Nr. 14199-803 beim Postscheckamt mit dem Vermerk »Markt & Technik, 64er« oder schicken Sie uns DM 5,— als Scheck oder in Bargeld. Der Verlag behält sich die Veröffentlichung längerer Texte vor. Kleinanzeigen, die entsprechend gekennzeichnet sind, oder deren Text auf eine gewerbliche Tätigkeit schließen läßt, werden in der Rubrik »Gewerbliche Kleinanzeigen« zum Preis von DM 11,— je Zeile Text veröffentlicht.

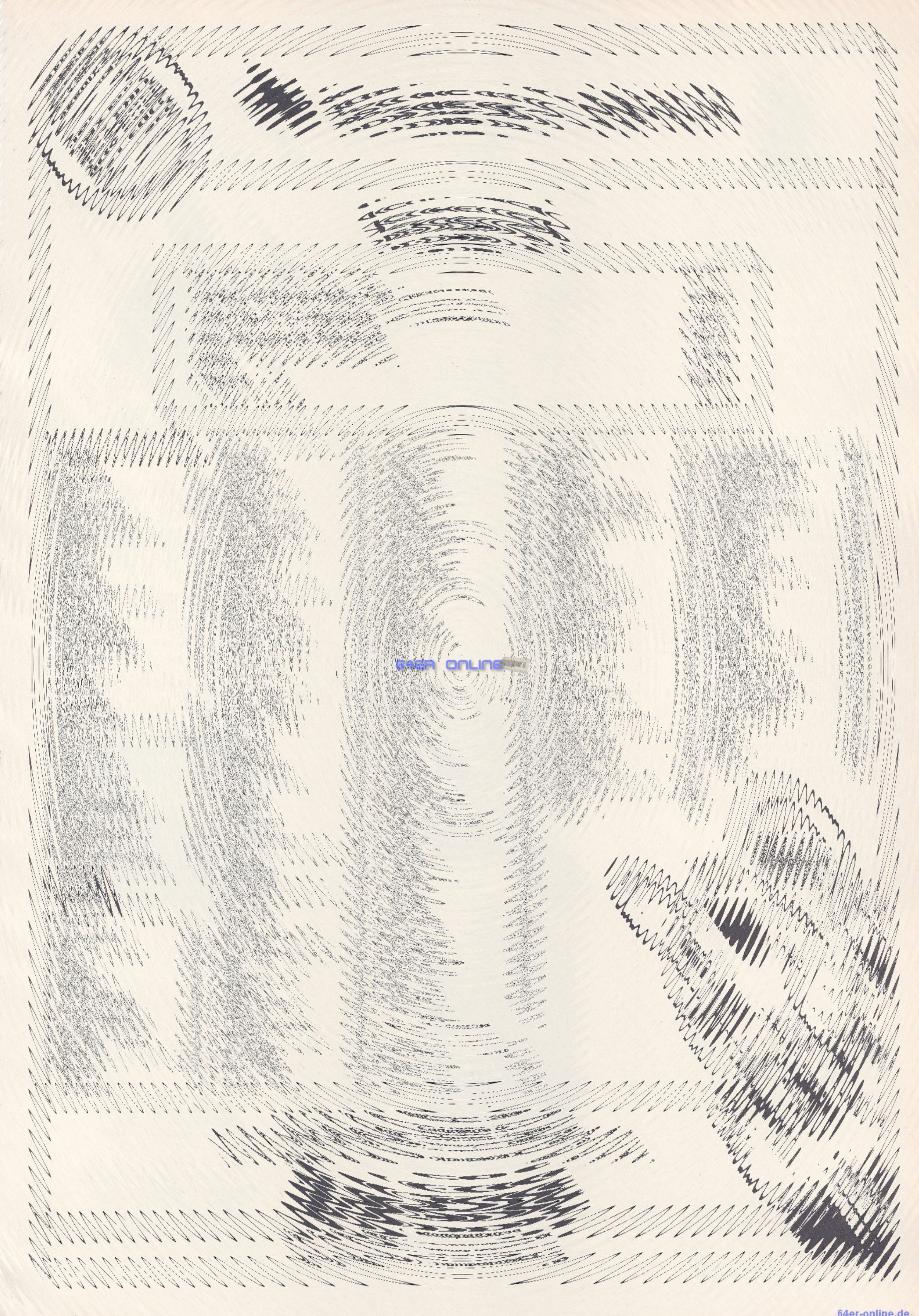
Private Kleinanzeigen

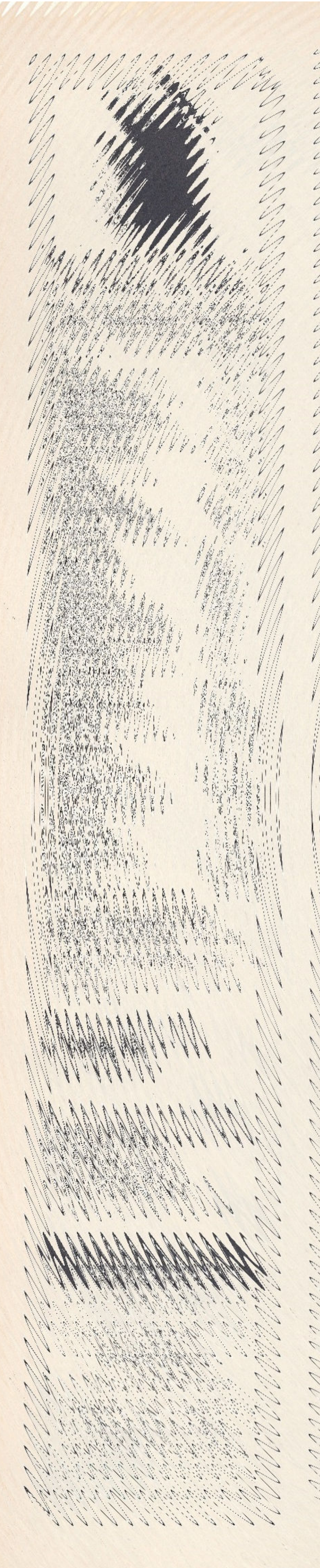
Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen



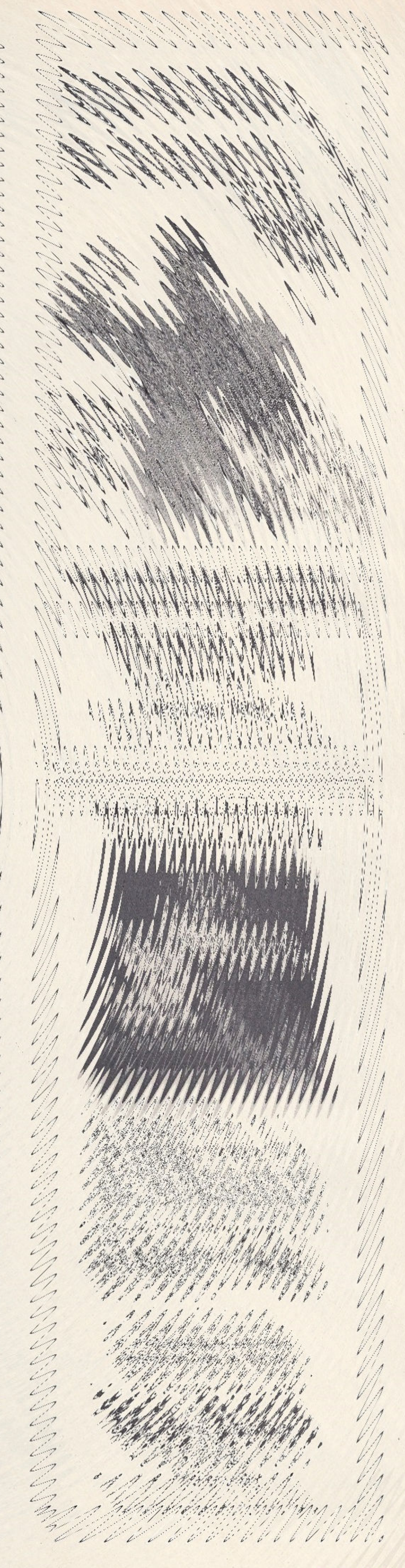






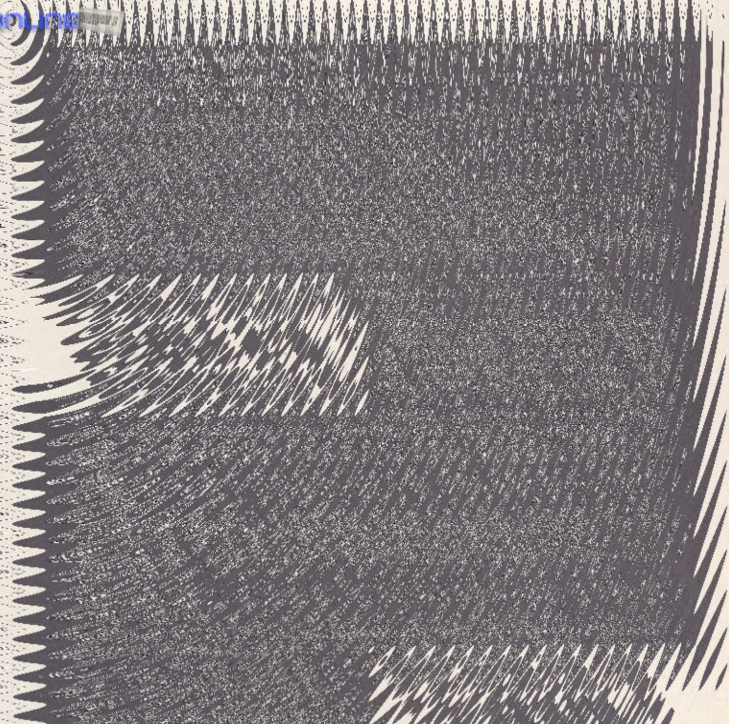


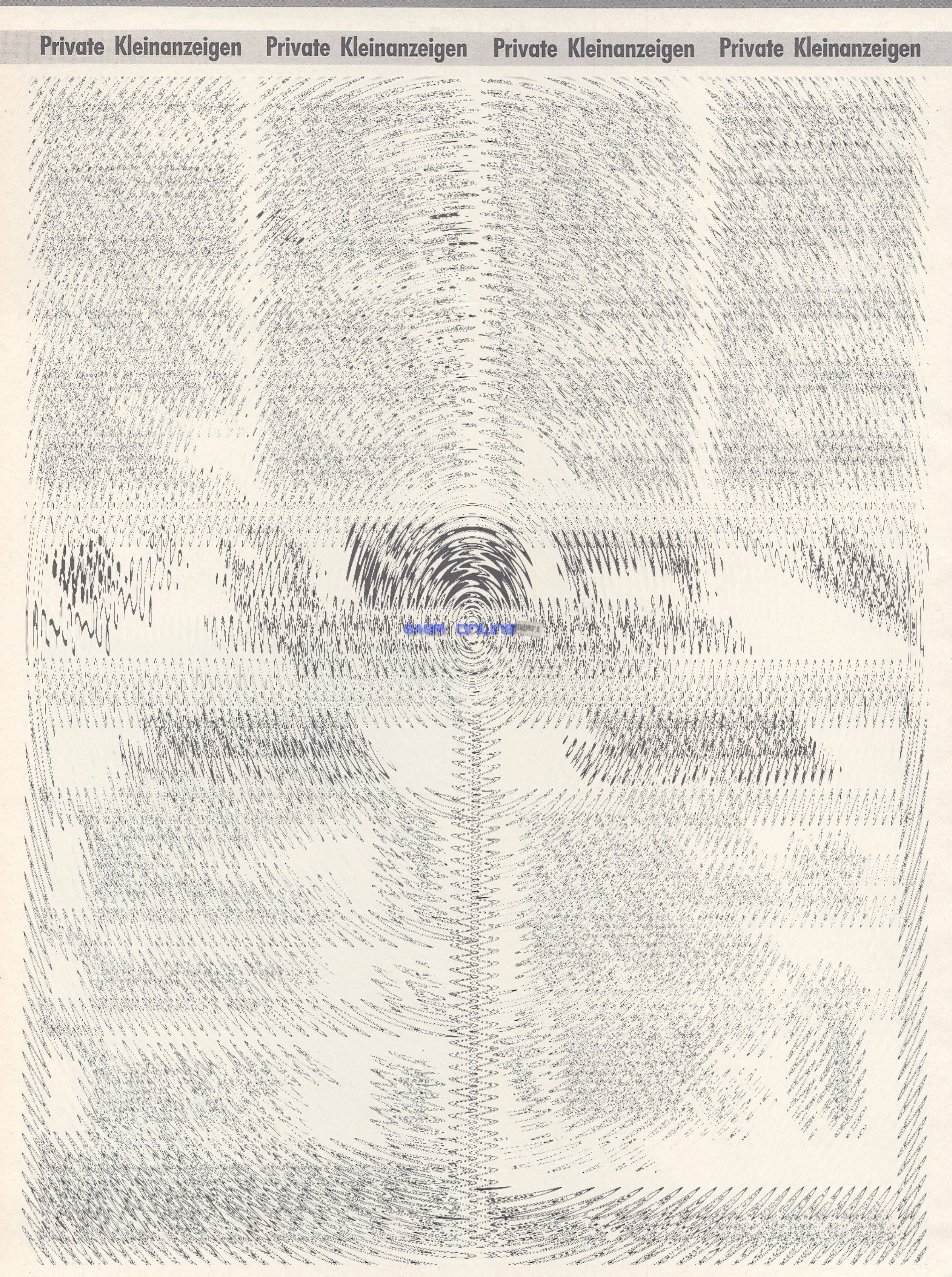




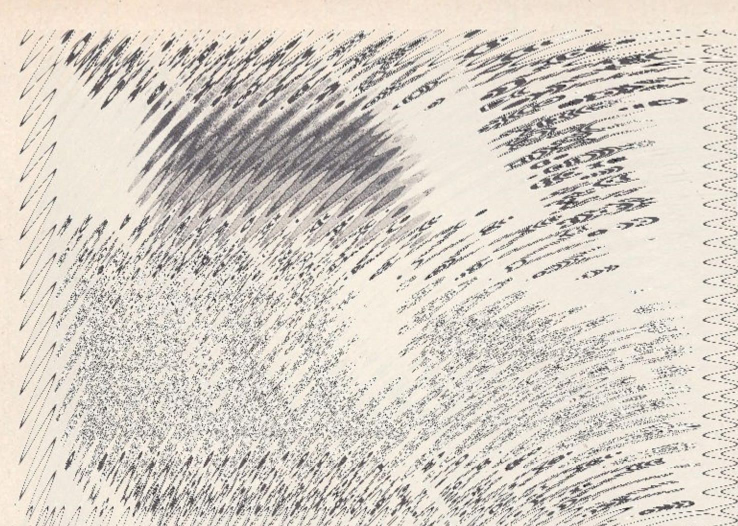
COMPUTER-MARKT

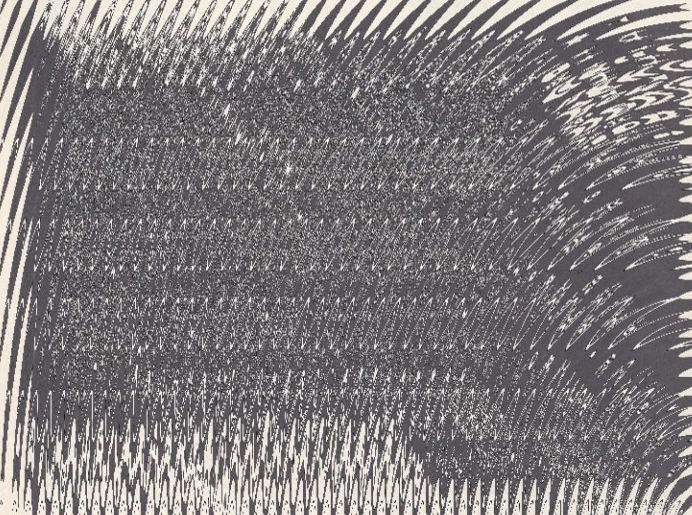
Private Kleinanzeigen Private Kleinanzeigen

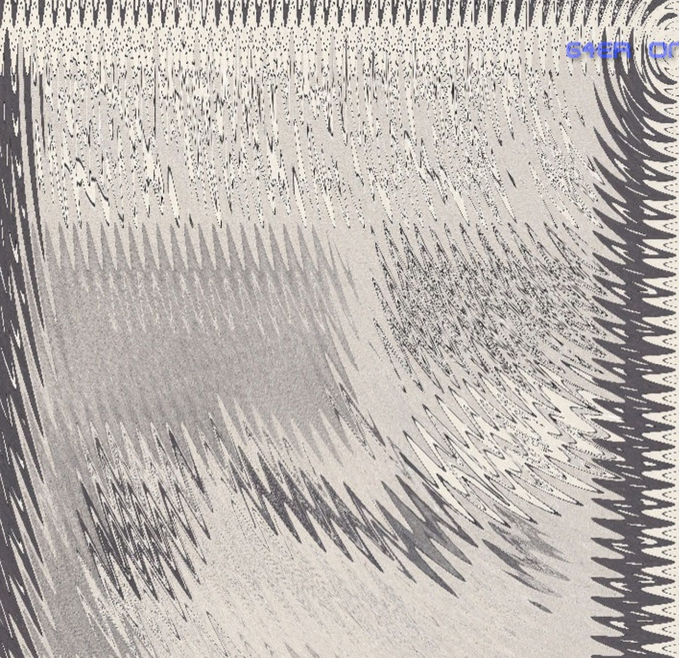




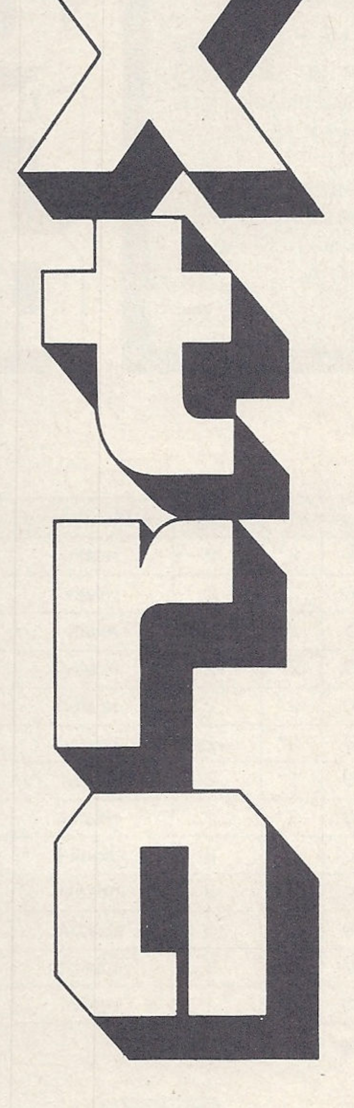












Ein Extra-Service der 64'er-Redaktion

Mit dem 64'er Extra stellen wir Ihnen einen neuen Service der 64'er-Redaktion vor: Zwei Extra-Seiten zum Heraustrennen und Sammeln in jeder Ausgabe. Um Mißverständnissen vorzubeugen: Das 64'er-Extra ist kein Kurs und kein Computer-Lehrgang, sondern ein Nachschlagewerk für all diejenigen, die ihren C 64 selbst programmieren.

In lockerer Folge wird das 64'er Extra Ihnen zukünftig geballte Informationen übersichtlich präsentieren. Die Extra-Seiten sind zum Heraustrennen und Sammeln konzipiert, und davon sollten Sie auch regen Gebrauch machen. Wir haben die Extra-Seiten bewußt flexibel gestaltet, damit Sie das 64'er Extra in der Ihnen am sinnvollsten erscheinenden Form archivieren können:

Sie können die Doppelseite komplett als Poster über Ihren Arbeitstisch an die Wand hängen, Sie können das Extra aber auch einmal in der Mitte falten und in einer Klarsichthülle unterbringen — griffbereit neben Ihrem Computer. Natürlich können Sie die Extra-Seiten aber auch lochen, abheften, archivieren — auf jeden Fall aber sollten Sie die Extras sammeln, denn mit der Zeit bekommen Sie so ein einzigartiges Archiv über Ihren Commodore zusammen.

Das 64'er Extra bringt in dieser ersten Ausgabe eine geordnete Übersicht über den Befehlssatz des 6502/6510-Prozessors. Damit hat es jetzt endlich ein Ende mit dem Blättern in Assembler- und Maschinensprache-Handbüchern. Trennen Sie die beiden Seiten aus dem 64'er-Magazin heraus, und schon haben Sie alle für den Maschinensprache-Programmierer wichtigen Daten auf einen Blick parat.

Für die nächsten Extra-Ausgaben sind unter anderem folgende Themen vorgesehen:

— Sprites und Grafik
— nützliche PEEKs und POKEs
— der VIC II-Chip
— die Kernal-Routinen
— der SID-Chip
— die Zero Page
— die 1541-Floppy
— Spiele-POKEs
— CIAs und User-Port
— Forth
— Pascal
— Comal
— Drucker-Vergleichstabelle
— Simons Basic
... und vieles andere mehr.

Natürlich werden wir auch andere Commodore-Computer, beispielsweise den neuen C 128, in gebührender Form berücksichtigen. Sie sehen schon, mit dem 64'er Extra bekommen Sie im Laufe der Zeit eine Spezial-Bibliothek über Ihren Computer, mit der sich sinnvoll arbeiten läßt.

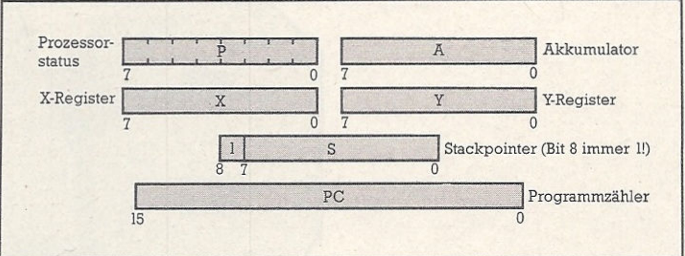
Selbstverständlich sind wir auch an Ihrer Meinung interessiert, denn das 64'er-Magazin ist die Computer-Zeitschrift zum Mitmachen. Wenn Sie Themenvorschläge für das 64'er Extra haben oder Ideen zur Gestaltung, dann schreiben Sie uns doch einfach mal eine Mitmachkarte. Sie können sicher sein, daß jeder Hinweis aufmerksam gelesen und ausgewertet wird. (ev)



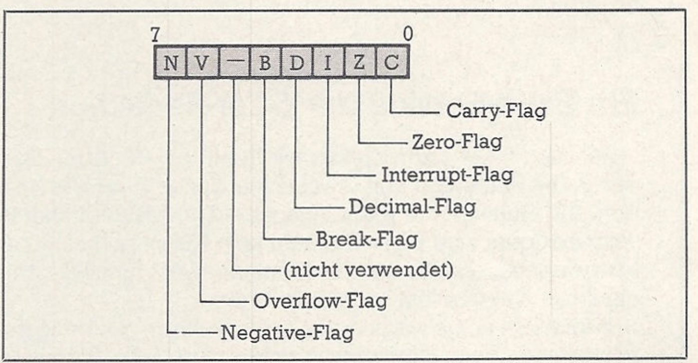
Das 64-er Extra ist eine Extra-Leistung für unseren Leser. Sie finden hier geballte Informationen für Ihren C 64 zum Herausnehmen und Sammeln.

In 64'er Extra 1 finden Sie eine geordnete Übersicht über den Befehlssatz des 6502/6520-Prozessors.

- Registersatz
- Flag-Register
- Alle Befehle auf einen Blick
- Alle Opcodes mit Flag-beeinflussung.



Registersatz des 6502/6510



Das Flag-Register

Befehl	op	n	Byte	Adressierung	Bedingung
BCC	90	2*	2	relativ	C = 0
BCS	b0	2*	2	relativ	C = 1
BEQ	f0	2*	2	relativ	Z = 1
BNE	d0	2*	2	relativ	Z = 0
BMI	30	2*	2	relaitv	N = 1
BPL	10	2*	2	relativ	N = 0
BVC	50	2*	2	relativ	V = 0
BVS	70	2*	2	relativ	V = 1
JMP	4c	3	3	absolut	keine
JMP	6c	5	3	indirekt	keine
JSR	20	6	3	absolut	keine
RTS	60	6	1	implizit	keine
RTI	40	6	1	implizit	keine

Sprung-Befehle

	Adressierungsarten						Flags				
	implizit (1 Byte)	immediate (2 Byte)	Zero Page (2 Byte)	absolut (3 Byte)	abs. X/Y (3 Byte)	Zero Page X/Y (2 Byte)					
Befehl	op n	op n	op n	op n	op n X/Y	op n X/Y	N	V	Z	C	Operation
CPX		e0 2	e4 3	ec 4			x	-	x	x	X—M
CPY		c0 2	c4 3	cc 4			x	-	x	x	Y—M
DEX	ca 2						x	-	x	-	X:=X—1
DEY	88 2						x	-	x	-	Y:=Y—1
INX	e8 2						x	-	x	-	X:=X+1
INY	c8 2						x	-	x	-	Y:=Y+1
LDX		a2 2	a6 3	ae 4	be 4* Y	b6 4 Y	x	-	x	-	X:=M
LDY		a0 2	a4 3	ac 4	bc 4 X	b4 4* X	x	-	x	-	Y:=M
STX			86 3	8e 4		96 4 Y	-	-	-	-	M:=X
STY			84 3	8c 4		94 4 X	-	-	-	-	M:=Y

Index-Register-Befehle

			Flags				
Befehl	op	n	N	V	Z	C	Operation
PHA	48	3	-	-	-	-	(S):=A; S:=S-1
PLA	68	4	x	-	x	-	S:=S+1; A:=(S)
PHP	08	3	-	-	-	-	(S):=P; S:=S-1
PLP	28	4	x	x	x	x	S:=S+1; P:=(S)
TAX	aa	2	x	-	x	-	X:=A
TXA	8a	2	x	-	x	-	A:=X
TAY	a8	2	x	-	x	-	Y:=A
TYA	98	2	x	-	x	-	A:=Y
TSX	ba	2	x	-	x	-	X:=S
TXS	9a	2	-	-	-	-	S:=X

Transfer-Befehle (1 Byte)

			Flags							
Befehl	op	n	N	V	B	D	I	Z	C	Operation
NOP	ea	2	-	-	-	-	-	-	-	keine
BRK	00	7	-	-	1	-	1	-	-	JMP ($FFFE)
CLC	18	2	-	-	-	-	-	-	0	C := 0
CLD	d8	2	-	-	-	0	-	-	-	D := 0
CLI	58	2	-	-	-	-	0	-	-	I := 0
CLV	b8	2	-	0	-	-	-	-	-	V := 0
SEC	38	2	-	-	-	-	-	-	1	C := 1
SED	f8	2	-	-	-	1	-	-	-	D := 1
SEI	78	2	-	-	-	-	1	-	-	I := 1

Spezial-Befehle (1 Byte)

	Adressierungsarten																						
	immediate		Zero Page		absolut		abs. X		abs. Y		(ind. X)		(ind. Y)		Zero Pg. X		Akku						
	(2 Byte)		(2 Byte)		(3 Byte)		(3 Byte)		(3 Byte)		(2 Byte)		(2 Byte)		(2 Byte)		(1 Byte)		Flags				
Bef.	op	n	op	n	op	n	op	n	op	n	op	n	op	n	op	n	op	n	N	V	Z	C	Operation
ADC	69	2	65	3	6d	4	7d	4*	79	4*	61	6	71	5*	75	4			x	x	x	x	A:=A+M+C
AND	29	2	25	3	2d	4	3d	4*	39	4*	21	6	31	5*	35	4			x	-	x	-	A:=A and M
ASL			06	5	0e	6	1e	7							16	6	0a	2	x	-	x	x	
BIT			24	3	2c	4													M_7	M_6	x	-	A and M
CMP	c9	2	c5	3	cd	4	dd	4*	d9	4*	c1	6	d1	5*	d5	4			x	-	x	x	A - M
	DEC				c6	5	ce	6	de	7					d6	6			x	-	x	-	M:= M - 1
EOR	49	2	45	3	4d	4	5d	4*	59	4*	41	6	51	5*	55	4			x	-	x	-	A:=A eor M
INC			e6	5	ee	6	fe	7							f6	6			x	-	x	-	M:=M + 1
LDA	a9	2	a5	3	ad	4	bd	4*	b9	4*	a1	6	b1	5*	b5	4			x	-	x	-	A:= M
LSR			46	5	4e	6	5e	7							56	6	4a	2	0	-	x	x	
ORA	09	2	05	3	0d	4	1d	4*	19	4*	01	6	11	5*	15	4			x	-	x	-	A:=A or M
ROL			26	5	2e	6	3e	7							36	6	2a	2	x	-	x	x	
ROR			66	5	6e	6	7e	7							76	6	6a	2	x	-	x	x	
SBC	e9	2	e5	3	ed	4	fd	4*	f9	4*	e1	6	f1	5*	f5	4			x	x	x	x	A:=A-M-1+C
STA			85	3	8d	4	9d	5	99	5	81	6	91	6	95	4			-	-	-	-	M:=A

Arithmetisch-Logische Befehle und Speicherzugriff

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	BRK	ORA ,X				ORA ZP	ASL ZP		PHP	ORA #	ASL A			ORA	ASL	
1	BPL	ORA ,Y				ORA ZP, X	ASL ZP, X		CLC	ORA ,Y				ORA ,X	ASL ,X	
2	JSR	AND ,X			BIT ZP	AND ZP	ROL ZP		PLP	AND #	ROL A		BIT	AND	ROL	
3	BMI	AND ,Y				AND ZP, X	ROL ZP, X		SEC	AND ,Y				AND ,X	ROL ,X	
4	RTI	EOR ,X				EOR ZP	LSR ZP		PHA	EOR #	LSR A		JMP	EOR	LSR	
5	BVC	EOR ,Y				EOR ZP, X	LSR ZP, X		CLI	EOR ,Y				EOR ,X	LSR ,X	
6	RTS	ADC ,X				ADC ZP	ROR ZP		PLA	ADC #	ROR A		JMP I	ADC	ROR	
7	BVS	ADC ,Y				ADC ZP, X	ROR ZP, X		SEI	ADC ,Y				ADC ,X	ROR ,X	
8		STA ,X			STY ZP	STA ZP	STX ZP		DEY		TXA		STY	STA	STX	
9	BCC	STA ,Y			STY ZP, X	STA ZP, X	STX ZP, Y		TYA	STA ,Y	TXS			STA ,X		
A	LDY #	LDA ,X	LDX #		LDY ZP	LDA ZP	LDX ZP		TAY	LDA #	TAX		LDY	LDA	LDX	
B	BCS	LDA ,Y			LDY ZP, X	LDA ZP, X	LDX ZP, Y		CLV	LDA ,Y	TSX		LDY ,X	LDA ,X	LDX ,Y	
C	CPY #	CMP ,X			CPY ZP	CMP ZP	DEC ZP		INY	CMP #	DEX		CPY	CMP	DEC	
D	BNE	CMP ,Y				CMD ZP, X	DEC ZP, X		CLD	CMP ,Y				CMP ,X	DEC ,X	
E	CPX #	SBC ,X			CPX ZP	SBC ZP	INC ZP		INX	SBC #	NOP		CPX	SBC	INC	
F	BEQ	SBC ,Y				SBC ZP, X	INC ZP, X		SED	SBC Y				SBC ,X	INC ,X	

Der 6502/6510-Befehlssatz auf einen Blick

64'er Extra 1

In dieser ersten Ausgabe des 64'er Extra erhalten Sie eine geordnete Übersicht über den Befehlssatz des 6502/6510-Prozessors. Jeder Befehl ist mit allen Adressierungsarten aufgelistet. Bei einigen Befehlen muß bei Überschreitung einer Seitengrenze ein zusätzlicher Taktzyklus (»*«) eingefügt werden. Für jede Adressierungsart sind der Opcode (op) und die Anzahl der benötigten Taktzyklen (n) einzeln aufgelistet. Die Länge der Instruktion in Bytes ist ebenfalls angegeben.

Sehr wichtig ist das Verhalten der Flags, die darum ebenfalls aufgeführt wurden. Die Bedeutung der einzelnen Flags:

Negative Flag (N): wird gesetzt, wenn das Ergebnis einer Operation negativ ist, sonst gelöscht.

Overflow Flag (V): wird gesetzt, wenn bei einer arithmetischen Operation ein Überlauf auftritt, andernfalls gelöscht.

Zero Flag (Z): wird gesetzt, wenn das Ergebnis einer Operation Null ist, andernfalls gelöscht.

Carry Flag (C): enthält den Übertrag vom höchstwertigen Bit bei arithmetischen Operationen und hat daneben spezielle Aufgaben bei Rotations- und Schiebebefehlen.

Dezimal Flag (D): bewirkt die Ausführung der Befehle ADC und SBC im Dezimal-Modus, falls es gesetzt ist, andernfalls werden die Befehle binär ausgeführt (Normalfall). Das Dezimal-Flag wird im Zusammenhang mit BCD-Operationen verwendet.

Break-Flag (B): wird gesetzt, wenn ein BRK-Befehl ausgeführt wurde.

Interrupt Flag (I): bei gesetztem Interrupt-Flag sind Unterbrechungen gesperrt.

Die Flags D, B und I werden nur durch spezielle Befehle gesetzt oder gelöscht und werden im allgemeinen nicht weiter beeinflußt. Aus diesem Grund sind bei den Befehlstabellen nur die Flags N, V, Z und C aufgeführt. In der Spalte Flags bedeutet:

»-« das Flag wird durch den Befehl nicht beeinflußt.

»x« das Flag wird durch den Befehl gesetzt oder gelöscht.

»0« das Flag wird durch den Befehl in jedem Fall gelöscht.

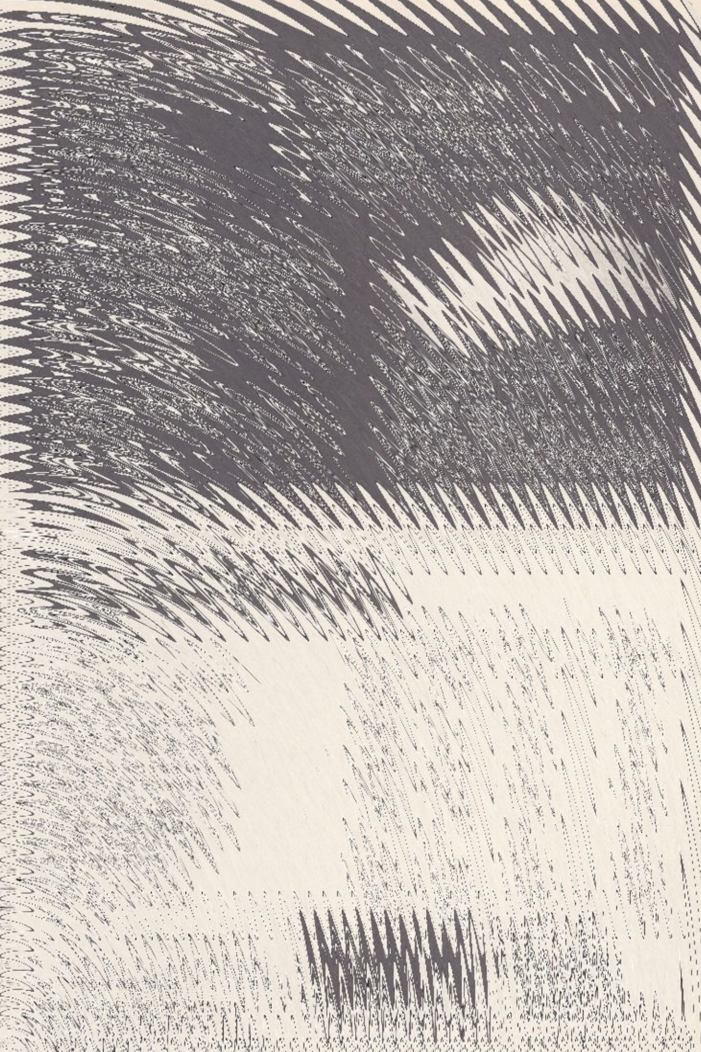
»1« das Flag wird durch den Befehl in jedem Fall gesetzt.

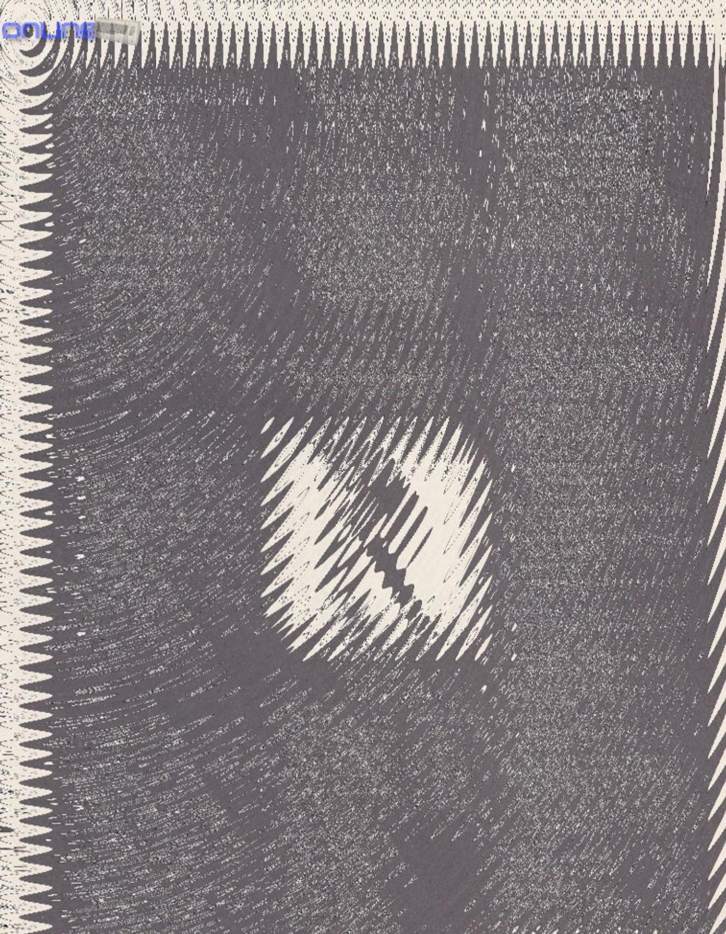
»M7« das Flag enthält den Zustand von Bit 7 der adressierten Speicherstelle.

»M6« das Flag enthält den Zustand von Bit 6 der adressierten Speicherstelle.

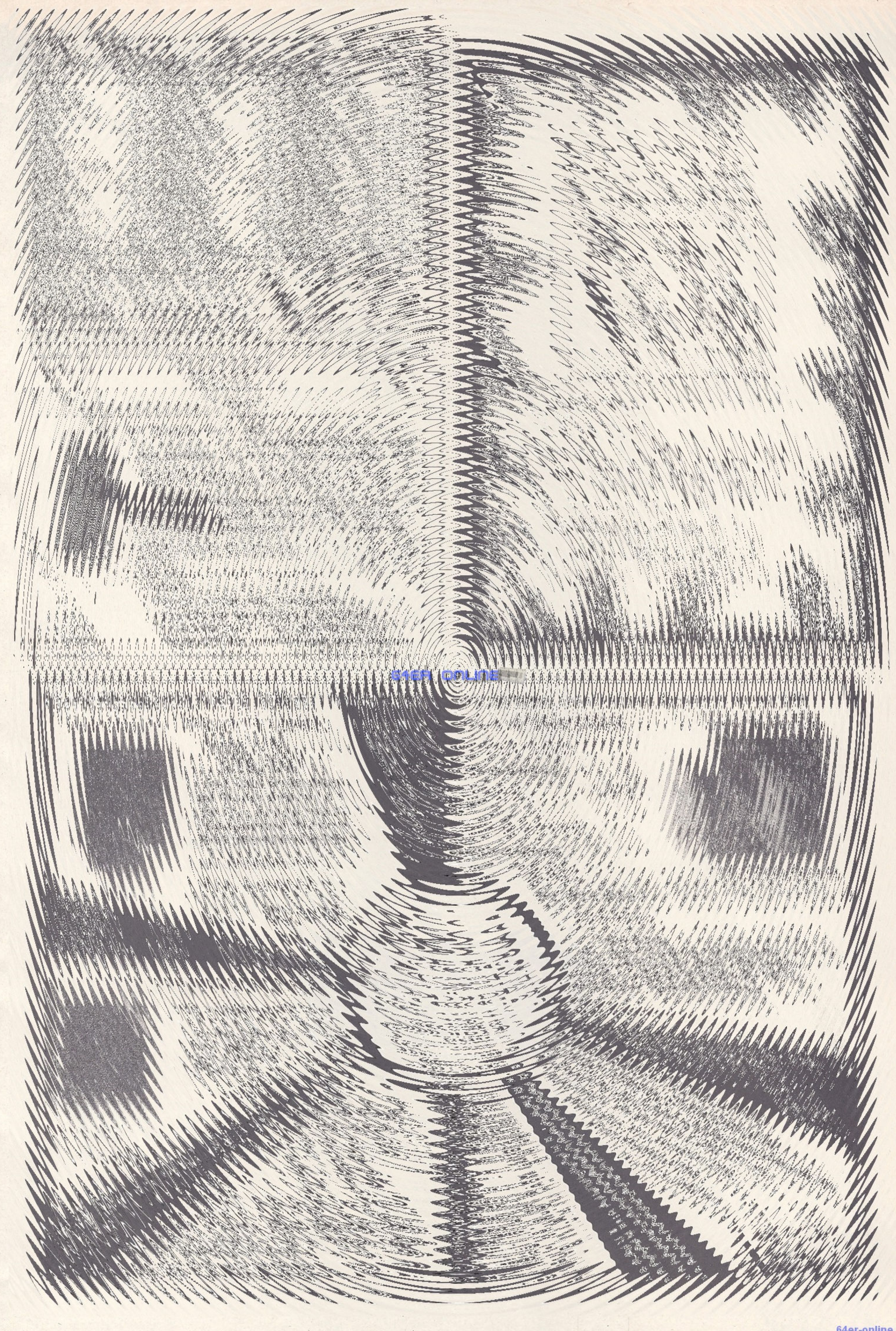
In der Spalte »Operation« bezeichnet A den Akkumulator, X das X-Register, Y das Y-Register, S den Stackpointer, (S) die vom Stackpointer adressierte Speicherstelle, C das Carry-Flag und M die durch den Befehl adressierte Speicherstelle.

Private Kleinanzeigen Private Kleinanzeigen

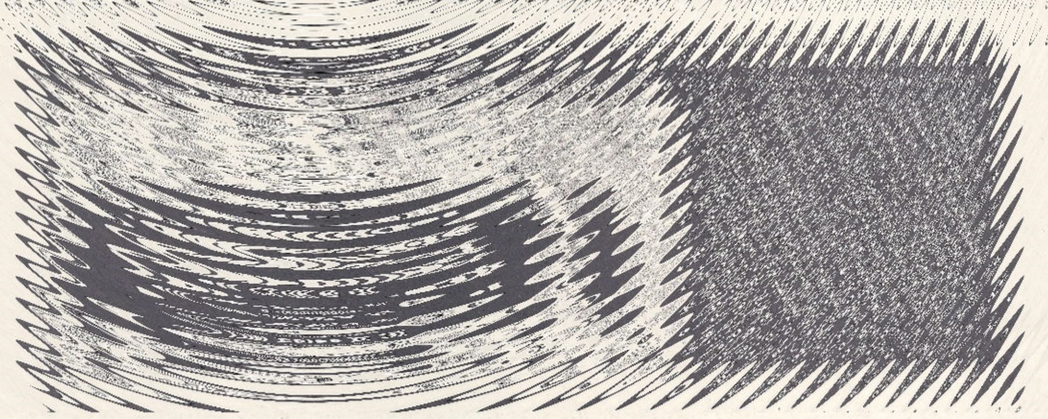


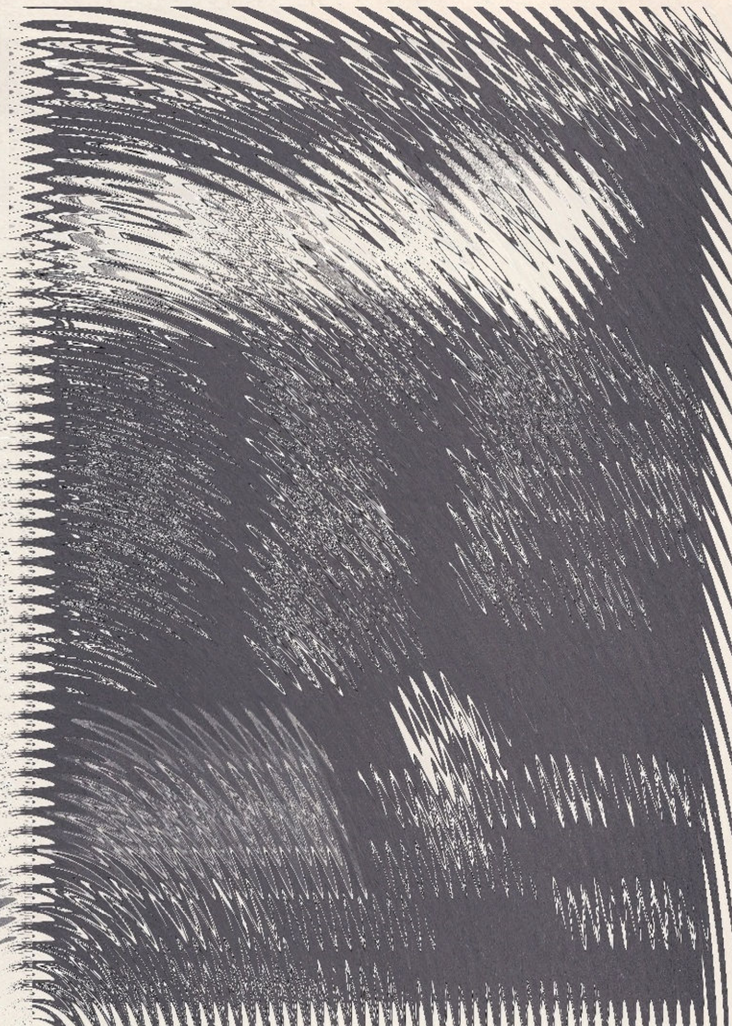


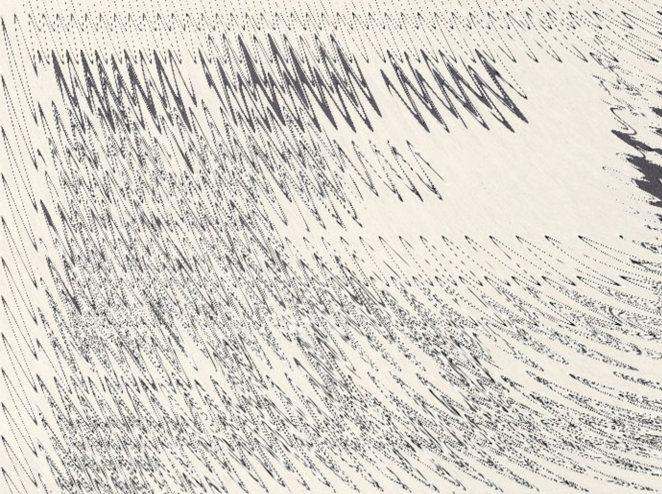










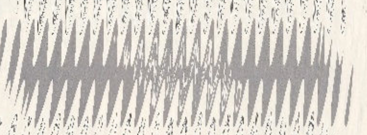


Private Kleinanzeigen

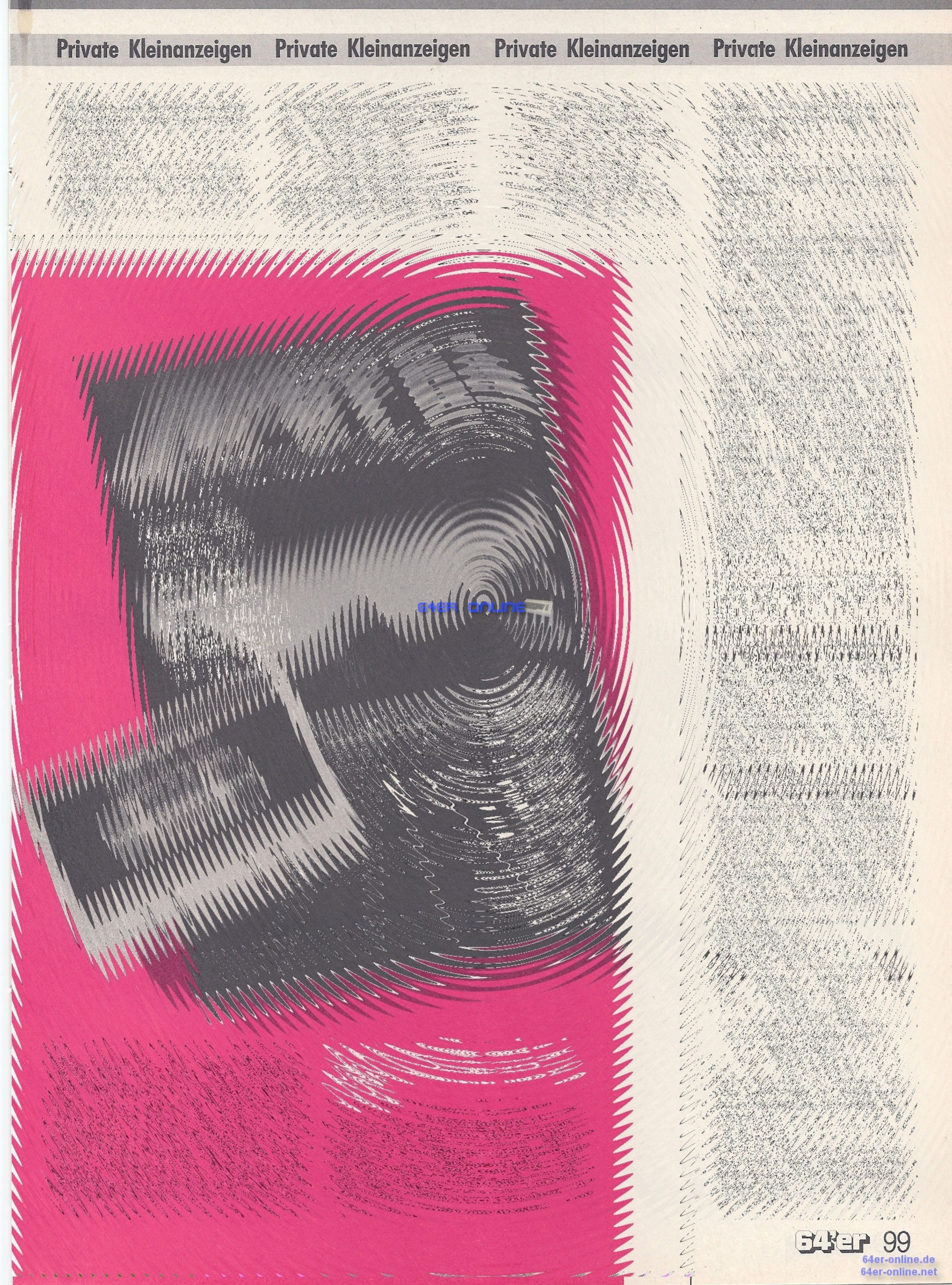


COMPUTER-MARKT

Private Kleinanzeigen Private Kleinanzeigen





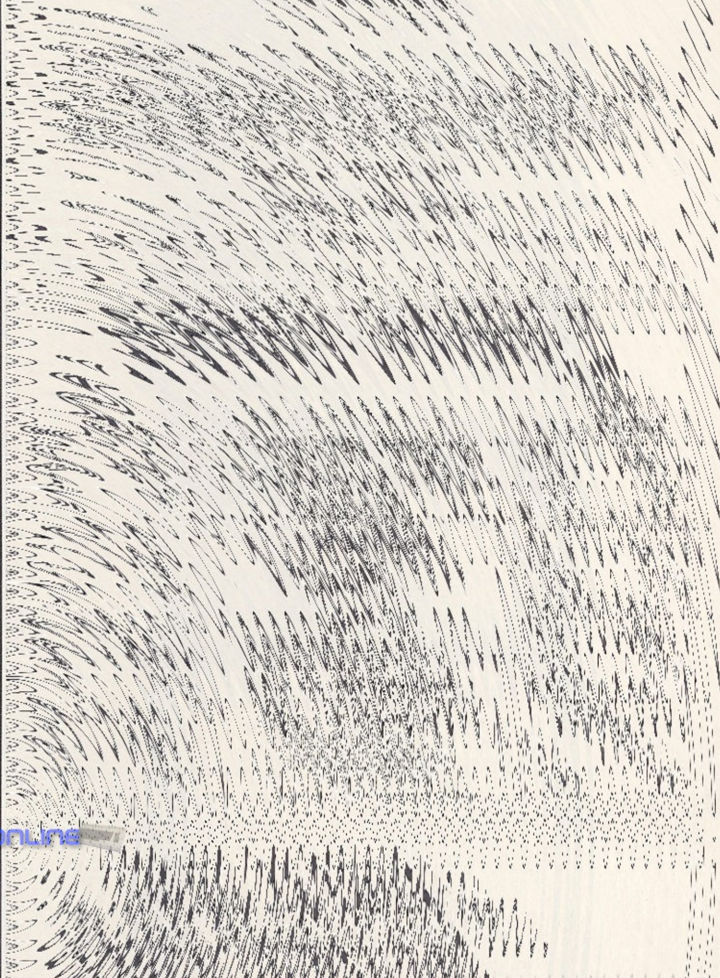


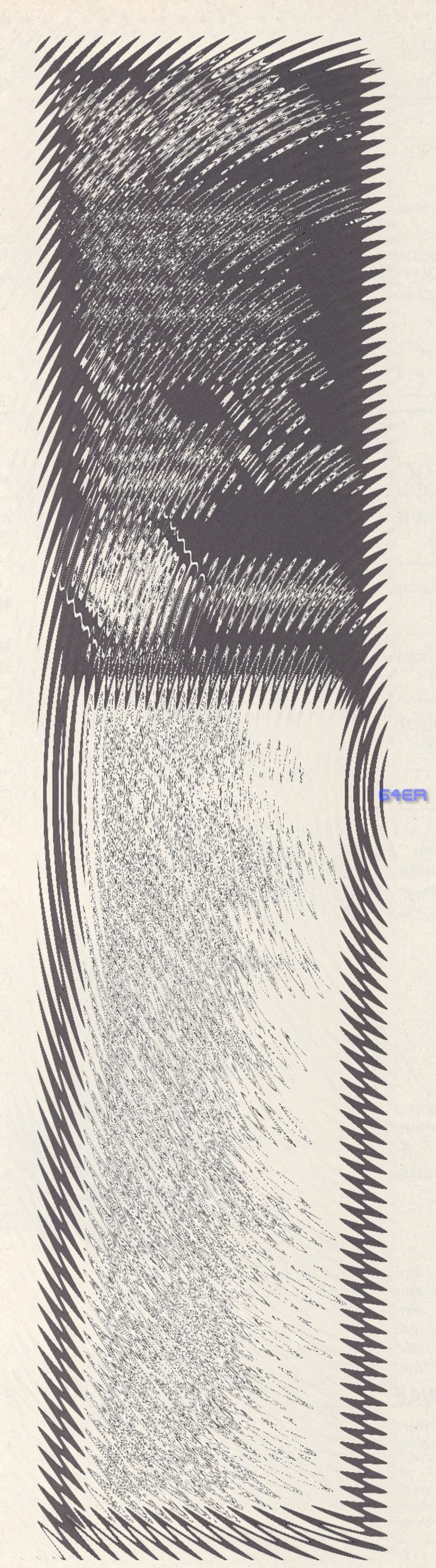
Private Kleinanzeigen   Private Kleinanzeigen

Gewerbliche Kleinanzeigen

COMPUTER-MARKT

Gewerbliche Kleinanzeigen





COMPUTER-MARKT

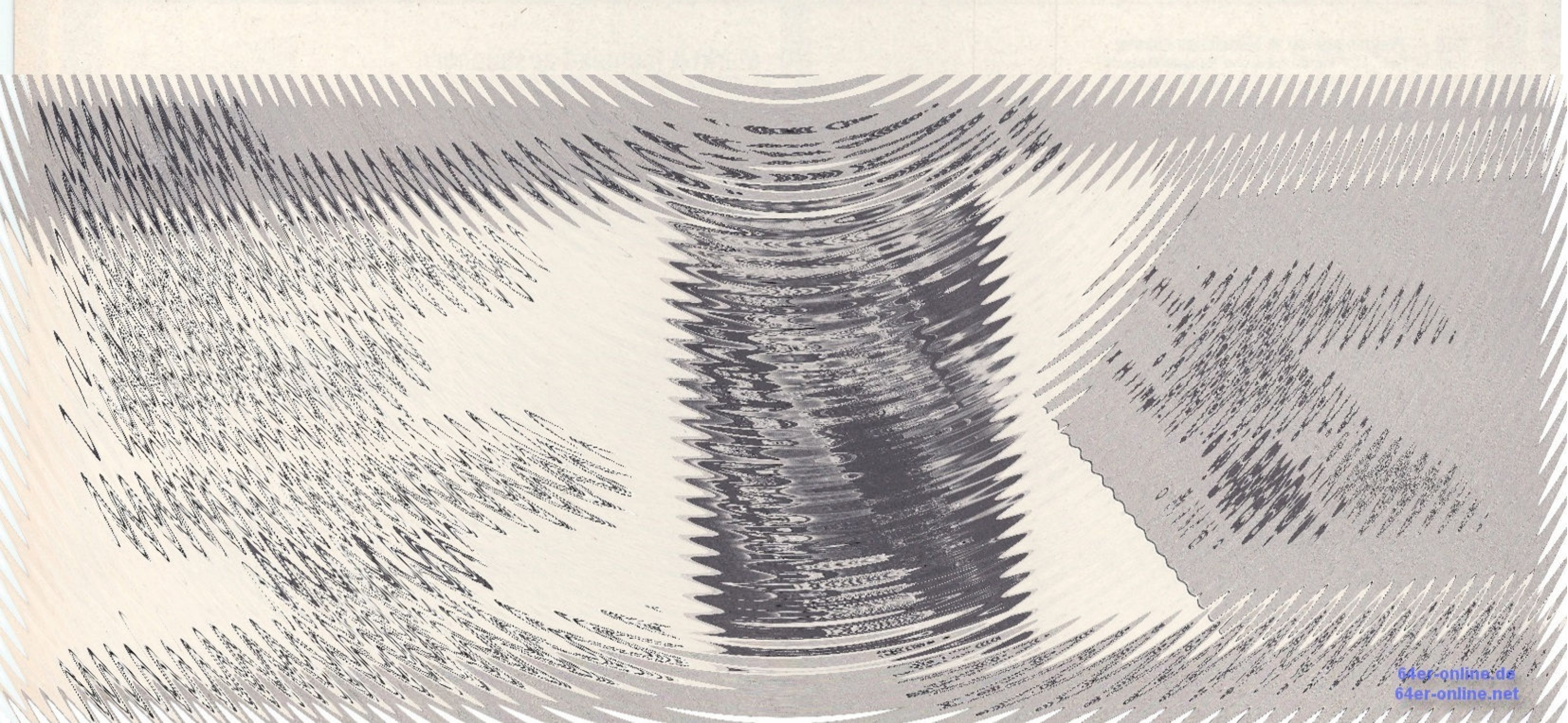
Gewerbliche Kleinanzeigen



Gewerbliche Kleinanzeigen

Gewerbliche Kleinanzeigen







Streifzüge durch die Grafik-Welt (Teil 1)

Grafik ist für viele Computer-Freunde ein wahres Zauberwort und ein fast unerschöpfliches Thema. Wir stellen Ihnen ein Grafiksystem vor, mit dem Sie in Zukunft alle Programme des Kurses mit den verschiedensten Grafik-Hilfen verwenden können.

Zwar haben wir in all den bisherigen Folgen des Grafik-Kurses auch immer ein paar Programmbeispiele gezeigt, besonders in den letzten waren wir aber reichlich theoretisch. Und weil Sie ja nun sicherlich Hires-3 mal ausprobieren möchten, sollen diesmal einige Beispiele gezeigt werden. Außerdem muß unbedingt mal etwas gegen die Sprachverwirrung in Fragen der »Computergrafik« getan werden! Deshalb unternehmen wir den Versuch, dieses Feld etwas genauer zu beschreiben.

1. Was ist eigentlich Computergrafik?

Alle Welt spricht von Computergrafik, aber kaum jemand erklärt, was man darunter eigentlich zu verstehen hat. Sie werden sagen, daß das doch ganz einfach sei: Computergrafik ist Grafik, die mittels des Computers gemacht wird. Das betrifft dann beispielsweise Sprites oder Shapes, Filmgrafik wie in TRON, bewegten Hintergrund wie bei bestimmten Spielen, Konstruktionsverfahren mittels CAD, Auswertung von Satellitenaufnahmen durch Falschfarbenverfahren, Balkendiagramme aus kaufmännischen Anwendungen, Simulation von Dreikörperproblemen in der Physik, Untersuchung von dreidimensionalen Funktionen in der Mathematik, Identifizierung von Objekten per Computer, und so weiter. Ich hoffe, Sie haben bemerkt, daß wir da ein enorm breites Sammelsurium von Computergrafik haben. Es ist dringend nötig, irgendwelche Unterscheidungen zu treffen, Strukturen festzustellen. Zunächst einmal, »Computergrafik« sollte besser als grafische Datenverarbeitung bezeichnet werden und betrifft dann drei mögliche Kombinationen von Daten und Grafik:

1) Zur Eingabe gelangen Daten, aus denen Bilder erzeugt werden. Das ist das, was wir künftig Grafik nennen werden. Dazu gehören dann beispielsweise Funktionenplots, Szenarios für Simulationen, etc.

2) Bilder werden eingegeben und auch Bilder wieder erzeugt. Das ist die Aufgabe der Bildverarbeitung. Dazu gehören dann die Techniken, unterschiedliche Grauwerte mittels verschiedener Farben zu differenzieren oder mit einer Videokamera aufgenommene Bilder in irgendeiner Weise zu verfremden.

3) Die Eingabe erfolgt als Bilder, die Ausgabe als Daten. Diesen Weg nennt man Bildanalyse und das ist ein aufstrebender Zweig beispielsweise der militärischen Satellitenerkundung. Dabei liegt die Aufgabenstellung darin, die Unzahl an Bildinformationen per Computer zu untersuchen auf bestimmte Objektklassen, zum Beispiel Schiffe oder ähnliches. Es ergeben sich sodann Daten, die weiterverarbeitet werden. Wie diese drei Gebiete zusammenhängen, zeigt Bild 1.

Es gibt viel Gemeinsames der drei Bereiche: Datenstrukturen, Speichern und Darstellen der Bilder erzeugen Aufgabenstellungen oder Lösungen, die identisch sein können. Ein Beispiel: In der Bildverarbeitung stellt sich häufig die Aufgabe, Konturen nachzuzeichnen. Andererseits erfordert die Grafik oft das Ausfüllen von umrandeten Flächen. Beide Problemstellungen sind entgegengesetzte Aspekte eines Problems und es läßt sich leicht verstehen, daß ein gemeinsames theoretisches Gerüst zur Lösung verwendet wird.

Nachdem wir diese grundsätzliche Unterteilung getroffen haben, können wir uns auf die Grafik selbst konzentrieren. Pavlidis [1] hat eine Unterscheidung von Bildern getroffen, die weniger mit dem Augenschein als vielmehr mit der Programmierung und Darstellungstechnik zu tun hat. Er teilt die Bilder in vier Klassen ein:

Klasse 1

Das sind Abbildungen mit einer vollen Grauskala oder Farbbilder. Ein gutes Beispiel bilden Fernsehaufnahmen. Die Darstellung erfolgt durch Integer-Werte, die als Bildelemente oder Pixels in Matrizen enthalten sind.

Klasse 2

Sogenannte »Bilevel-Bilder«, weil sie nur zwei Zustände kennen: Schwarz und Weiß. Ein Beispiel kann eine Textseite in einem Buch sein. Die Bildinformationen sind als Bits in einer Matrix enthalten.

Klasse 3

Hiermit sind Kurven und Linien, also kontinuierliche Punktfolgen erfaßt. Beispiele bilden Umrisse von Objekten oder Funktionsgraphen. Die Bildspeicherung kann auf verschiedene Weise erfolgen: Durch Erfassen der X-Y-Koordinaten oder durch Verkettungscodes etc.

Klasse 4

Gruppen diskreter Punkte, die zu weit auseinander liegen, um noch als Kurven beschrieben zu werden. Die Erfassung geschieht meist in Variablen-Feldern (Arrays).

Wie Sie unschwer erkennen werden, ist die Trennung der einzelnen Klassen nicht sehr scharf. Besonders die Grenzen zwischen Klasse 1 und Klasse 2 sowie zwischen Klasse 3 und Klasse 4 sind fließend. Bedingt durch technische Anforderungen, werden häufig Bilder einer bestimmten Klasse durch andere Klassen realisiert. Zum Verständnis sollen zwei Beispiele dienen:

Flächen höherer Ordnung sind eigentlich Klasse 1- oder 2-Abbildungen. Trotzdem werden sie oft als Klasse 3-Bilder, nämlich als eine Anzahl von Kurvenzügen dargestellt.

Hardwaremäßig ist der Commodore 64 auf Bilder der Klassen 1 und 2 zugeschnitten. Kurvenplots, die eigentlich Klasse 3-Bilder sind, müssen als Klasse 2-Abbildungen umschrieben werden. (Dazu kommen wir gleich noch)

Transformationen sind Übergänge zwischen verschiedenen Bezugssystemen. Ein Beispiel wäre die Transformation, die in Bild 2 anhand eines Überganges von einem Koordinatensystem zu einem anderen gezeigt wird.

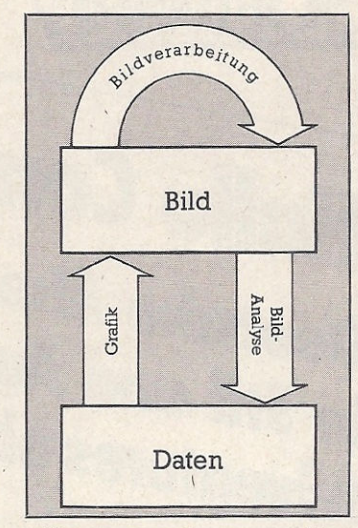


Bild 1. Das ist »Computergrafik«

Auch zwischen den vier Klassen gibt es Möglichkeiten der Transformation, die in der grafischen Datenverarbeitung mit bestimmten Namen versehen sind:

Übergang zwischen den Klassen

von 1 zu 2

Segmentation: Bildflächen werden in Bereiche etwa gleicher Grauwerte oder Helligkeitswerte gruppiert;

von 2 zu 3

Ermitteln von Konturen oder des Skelettes einer Region;

von 3 zu 4

Kurvensegmentation: Entlang einer Kurve werden charakteristische (oder kritische) Punkte gesucht. Das sind zum Beispiel die Ecken eines Polygons;

von 4 zu 3

Da gibt es zwei wichtige Begriffe:

Interpolation: Punkte werden durch Kurvenzüge verbunden.

Approximation: Kurvenanpassung an Punktgruppen unter minimaler Abweichung;

von 3 zu 2

Ausfüllen von Konturen oder Schattieren;

von 2 zu 1

Techniken zur Anreicherung der Bildinformationen.

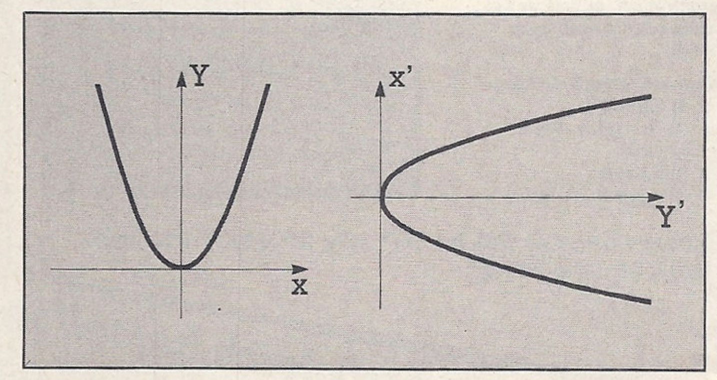


Bild 2. Beispiele für Transformationen

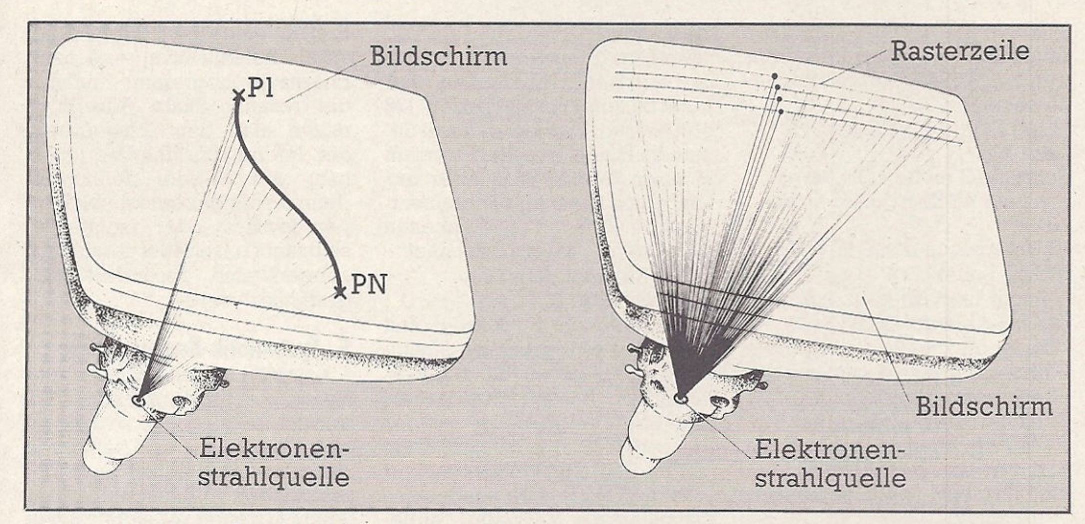


Bild 3. Bildschirmaufbau bei Vektorgrafik und Rastergrafik

Auch innerhalb der Klassen sind Transformationen von großer Bedeutung, sie werden uns noch häufig beschäftigen. Zunächst einmal gehören dazu Übergänge vom Räumlichen ins Flächenhafte: die sogenannten Projektionen. Der umgekehrte Weg wird übrigens Rekonstruktion genannt. Eng mit den Projektionen hängen die Probleme der Hinterschneidung zusammen, die wir im Grafik-Kurs, Folge 7 (64'er, Ausgabe 10/84) schon gestreift haben. Im englischen Sprachgebrauch unterscheidet man noch zwischen der »hidden line« und »hidden surface«- Problematik, was bedeutet »verborgene Linie« oder »verborgene Oberfläche«. Zu Transformationen innerhalb einer Klasse gehören auch Koordinatentransformationen, wie Drehung, Translation, Zerrung oder Stauchung von Systemen und natürlich die Übergänge zwischen verschiedenen Arten der Koordinatensysteme, wie kartesisch zu polar oder zylindrisch.

Bevor wir nun mit der Systematik der grafischen Datenverarbeitung weitermachen, brauchen wir noch einige Details über Bildeingabe und Bildwiedergabe.

2. Bildeingabe

Hauptsächlich bei der Bildverarbeitung und der Bilderkennung stellt sich das Problem der Bildeingabe. Wenn man allerdings den Grad der Interaktion bei der Grafikerstellung erhöht, sind auch dort Eingaben von Bildern denkbar. Zur Begriffsklärung: Interaktion nennt man das »Gespräch« zwischen Computer und Benutzer. So gibt es Programme, die auf einen Tastendruck ein fertiges Bild erstellen, ohne daß man — außer einem RUN/STOP — noch eine Eingriffsmöglichkeit hat. Andere erfragen ständig neue Eingaben, wie zum Beispiel die Joystickgrafik mittels des Programmes Hi-Eddi (Listing des Monats, 64'er, Ausgabe 1/85). Im ersten Fall ist der Grad der Interaktion gleich Null, im zweiten Fall ist er sehr hoch. Bei CAD-Anwendungen liegt ebenfalls ein hoher Interaktionsgrad vor und dort ist es bisweilen sinnvoll, ganze Bildteile nicht als Daten, sondern als Bilder einzugeben.

In jedem Fall stellt sich das Problem, ein analoges Bild in ein Feld diskreter Daten umzusetzen, es zu digitalisieren. Der Prozeß des Digitalisierens besteht aus zwei Teilen, nämlich dem Segmentieren und dem Quantisieren der Abbildung. Segmentieren: Die gesamte Bildfläche muß in Bildpunkte (möglichst charakteristische) unterteilt werden. Man spricht hier von »Raumauflösung« des Bildes.

Quantisieren: Jedem so gewonnenen Bildpunkt wohnt eine Bildcharakteristik inne (zum Beispiel Grauwert), die nun als Zahl ausgedrückt wird. Man spricht von der »Grauwert- (oder auch Farbwert-) Auflösung.

Schwerwiegende Probleme gibt es eigentlich nicht beim Digitalisieren. Man könnte beispielsweise ohne weiteres eine Fernsehkamera einsetzen, die jedes Objekt in eine ganze Anzahl von Punktzeilen rastert — also eine hohe Raumauflösung liefert — und auch zwischen Weiß und Schwarz sehr viele Grauabstufungen erfaßt — also eine gute Grauwertauflösung erzeugt. Die Schwierigkeit liegt eher in der genügend schnellen Verarbeitung der dadurch erzeugten Informationsflut. Deshalb verwendet man im allgemeinen lieber spezielle Bilderfassungsgeräte, sogenannte Digitizer, die in ihrem Verfahren der jeweiligen Computergeschwindigkeit eher angepaßt sind. Mit Hilfe sogenannter »Scanner« (das heißt »Abtaster«) wird ein Bild in genau definierte Zeilen oder sonstige Punktfolgen zerlegt und dann quantisiert.

Ein anderes Verfahren sind die Tablett-Digitizer, wo beispielsweise die Wechselwirkung zwischen einer magnetisierten Oberfläche und einem elektrisch geladenen Stift ausgenutzt wird. Für die interaktive grafische Datenverarbeitung gewinnen diese Geräte mehr und mehr an Bedeutung.

3. Bildwiedergabe

Die Wiedergabe von Daten als Bild ist komplexer als die Bildeingabe. Hier muß nicht nur die Umsetzung digitaler in analoge Signale stattfinden, sondern auch noch eine Anpassung an die jeweiligen Hardware-Gegebenheiten. Bei diesen unterscheidet man von vornherein zwei: Hardcopy-Ausgabe und CRT-Ausgabe. (CRT = »cathode-ray-tube«, auf deutsch: Bildschirm). Eine Hardcopy erfordert lediglich die Anpassung der Daten an geräteabhängige Formate und die Einhaltung einer gewissen Reihenfolge der Datenübermittlung.

Die Bildschirmdarstellung verlangt das erstere auch. Die Besonderheit bei dieser Art der Bildwiedergabe ist aber, daß die gesamte Abbildung mindestens 30mal pro Sekunde an das Bildschirmgerät gesandt werden muß. Ein Bild entsteht ja durch die Wechselwirkung eines Kathodenstrahles mit einer phosphoreszierenden Schicht auf der Bildschirmplatte. Je nach Intensität des auftreffenden Strahles wird ein mehr oder weniger helles Aufleuchten eines Bildpunktes bewirkt. Das Nachleuchten dieser Schicht ist nur recht kurz, so daß der Bildinhalt häufiger aufgefrischt werden muß. Das menschliche Auge nimmt weniger als 25—30 Auffrischungen als Flackern wahr, daher also die Mindestzahl von 30 kompletten Bildübermittlungen pro Sekunde.

Für jeden Bildpunkt müssen mindestens drei Informationen an das Bildschirmgerät gesandt werden: Die X- und die Y-Koordinate, sowie die Intensität (also der Grauwert) des Strahles an der so spezifizierten Stelle. Bei Farbbildern sind es sogar fünf Informationen, denn anstelle des Grauwertes muß für jede Grundfarbe (drei davon gibt es) ein Wert übermittelt werden. Die Mischung dieser Farbinformationen führt dann zur jeweiligen Farbdarstellung. All dies nimmt uns beim Commodore 64 glücklicherweise der VIC-II-Chip in Zusammenarbeit mit den CIAs ab. Das ist zwar einerseits eine gewaltige Erleichterung für uns Benutzer, führt aber andererseits auch — wie wir gleich sehen werden — zu gewissen Einschränkungen.

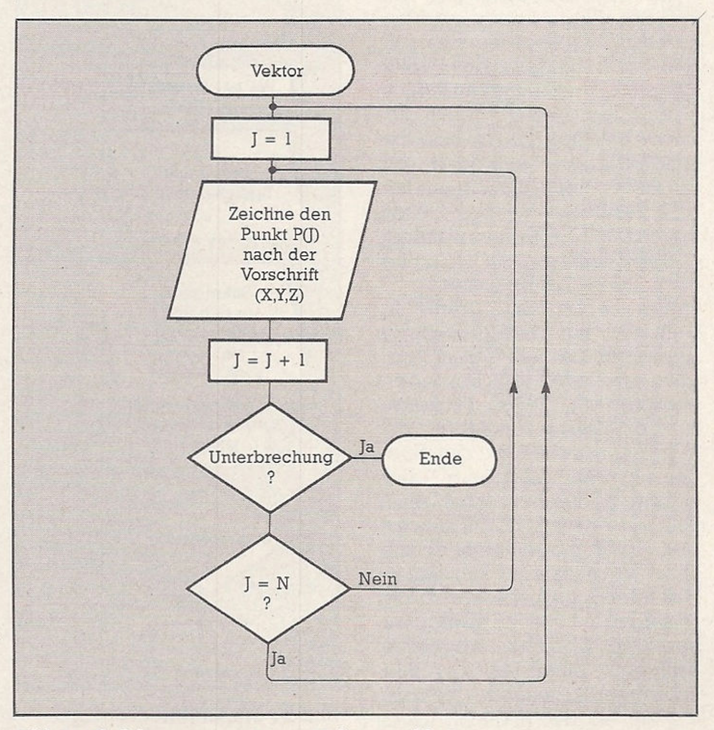


Bild 4. Flußdiagramm eines »Vektorgrafik-Programmes«



Man unterscheidet nämlich grundsätzlich zwei Verfahren der Bildwiedergabe per CRT-Gerät: Die Raster- und die Vektor-Methode. Sehen wir uns zunächst mal die Vektormethode an. Sie ist speziell für Bilder der Klassen 3 und 4 geeignet. Der Elektronenstrahl wird hier genau zum Punkt geführt, der durch die X- und Y-Koordinaten charakterisiert ist und an dieser Stelle dann einen Leuchtpunkt der gewünschten Intensität erzeugt. Alle anderen möglichen Bildschirmpositionen bleiben unberücksichtigt. Nehmen wir mal an, eine Abbildung setze sich zusammen aus der Punktfolge P1, P2, P3, ..., Pn (siehe auch Bild 3).

Als Programm könnte man die Arbeit der Bildwiederholung dann etwa so schreiben:

```
I = 1
Label
  Zeichne Punkt P(I) nach der Vorschrift X,Y,Z (dabei Z als Grauwert)
I = I + 1
IF Unterbrechung THEN END
IF J = N THEN J = 1
GOTO Label
```

Bild 4 zeigt einen Programmablaufplan dieser Routine. Was man gut erkennen kann: Der Elektronenstrahl durchläuft nur die Punktfolge. Will man einen Punkt entfernen oder ändern oder dazupacken, ist das ohne weitere Berücksichtigung aller anderen Punkte möglich. Die Bildwiederholung kann sehr schnell geschehen und ist abhängig von der Anzahl der zu bearbeitenden Punkte. Da sieht man dann auch gleich den Nachteil dieser Methode: Wenn die Abbildung sehr komplex wird, also beispielsweise ein Bild der Klasse 1 oder 2 auf diese Weise erzeugt werden soll, dann wird die Reihe der Punkte P(I) ziemlich lang und es kann in Grenzfällen zum Flackern des Bildes kommen. Der Vorzug dieses Verfahrens liegt darin, daß die Auflösung praktisch nur von den gerätetechnischen Details des Bildschirmgerätes abhängt. Übliche Auflösungen liegen bei etwa 4096x4096 Bildpunkten.

Was ist nun Rastergrafik? Jedem Ort auf dem Bildschirm entspricht hier ein Ort im Speicher des Computers. Besonders geeignet ist dieses Verfahren zur Darstellung von Bildern der Klassen 1 und 2. Aus der Speicheradresse folgt der Bildschirmort, aus dem Inhalt einer solchen Adresse der Grauwert, den der Kathodenstrahl an diesem Ort realisieren soll. Bei jeder Bildwiederholung wird der gesamte für diesen Zweck reservierte Speicher abgetastet und sein Inhalt an das Bildschirmgerät gesandt. Nehmen wir mal an, der Bildschirmspeicher sei 8000 Byte groß (N= 8000), dann könnte man hier ebenfalls ein Programm schreiben zur Bildwiederholung (siehe auch Bild 3).

```
Label FOR I = 1 TO 8000
Lese Z
Schreibe Z an den Ort, der sich aus I als X-Y-Koordinate ergibt
NEXT I
IF Unterbrechung THEN END
GOTO Label
```

Auch hier wieder ein Programm-Ablaufplan (Bild 5):

Es ist gleichgültig, ob nur ein einziger Punkt zu setzen ist oder alle: Jedesmal wird der gesamte Speicherinhalt übertragen, die Ausführungszeit ist also konstant und nur von der Auflösung — hier also der Speichergröße N — abhängig.

Der Umgang mit Rastergrafik ist komplizierter als der mit Vektorgrafik. Wenn sich beispielsweise zwei Abbildungsbereiche überlappen, darf man beim Rasterverfahren nicht einfach einen davon löschen, denn damit wirkt man auch auf die an derselben Stelle liegenden Teile des zweiten Bereiches ein. Die Vektorgrafik nimmt einfach die Punkte aus dem Auffrischungszyklus heraus, die zu löschen sind. Weil bei dem Rasterverfahren zwischen Bildschirm und Speicher ein 1:1-Verhältnis besteht, kommt man ziemlich schnell an die Grenzen der möglichen Auflösung. Schon eine Auflösung von 1024*1024 Pixels erfordert bei Klasse 2-Bildern (also mit einer Bit-Map) 128 KBytes Speicherraum. Nun werden Sie sagen, daß Speicherraum ja immer preiswerter zu haben ist und man deshalb ohne weiteres zu guten Auflösungen, die denen

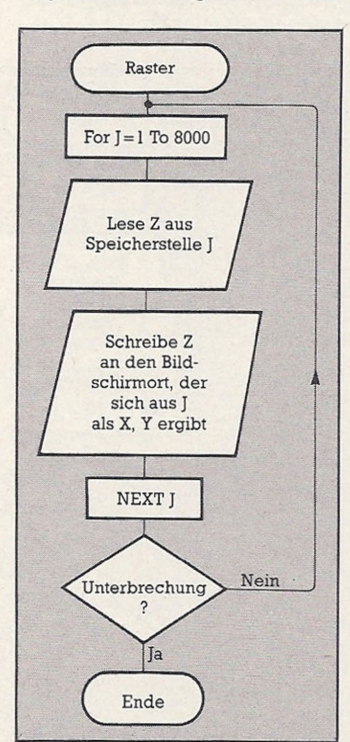


Bild 5. Bildschirmaufbau bei Rastergrafik

des Vektorverfahrens gleichkommen, gelangen könnte. Dabei wird aber übersehen, daß zum Beispiel diese ganzen 128 KByte innerhalb einer 30stel Sekunde durchgeblättert werden müssen. Nimmt man aber nur mal einen Lese- und Schreibvorgang — ohne auf die Feinheiten zu achten — wie zum Beispiel:

```
LDA (Adresse 1),Y
STA (Adresse 2),Y
```

und sieht sich an, wieviel Zeit diese Sequenz braucht (nämlich etwa 11 Taktzyklen für ein Byte an Informationen), dann dauert das Lesen und Übertragen der ganzen 128 KByte immerhin schon 1 441 792 Taktzyklen und das muß auch noch mindestens 30mal in der Sekunde passieren! Außer einem großen Speicher ist somit auch ein sehr schneller Mikroprozessor vonnöten und genau da liegen die Grenzen dieses Verfahrens.

4. Standortbestimmungen

Nach all diesen Details können wir schonmal eine Standortbestimmung wagen: Unser Commodore 64 ist eingerichtet auf Rastergrafik. Er kann mit Hilfe des VIC-II-Chip den Inhalt einer 8000 Byte großen Bit-Map auf dem Bildschirm zeigen. Der 3. Wert, also der Grauwert Z, kommt aus einem weiteren 1000 Byte großen Speicher, wie Ihnen noch aus den ersten Folgen dieser Serie in Erinnerung sein dürfte. Der zu einer bestimmten Bildschirmposition gehörige Bitwert der Bit-Map sagt bei uns nur aus, ob das obere oder das untere Nibble des Grau- (oder Farb-) Codes aus dem 1000 Byte großen Speicher realisiert werden soll. Noch komplizierter ist der Multicolormodus, der uns den Übergang zu Klasse 1-Abbildungen ermöglicht.

Einige weitere Standortbestimmungen betreffen Hires-3 und unsere weiteren Grafik-Folgen. Wir werden uns nahezu ausschließlich der Grafik im engeren Sinn widmen, also der Wiedergabe von Daten als Bilder. Wegen der Auslegung von HiRes-3 und der besseren Auflösung beschränken wir uns auf Klasse 2 Abbildungen, wobei wir Klasse 3 und 4 nachbilden können bzw. auch Transformationen von Klasse 4 nach Klasse 3 und Klasse 2 durchführen werden. Transformationen innerhalb der Klassen werden in allen Formen auftreten (Projektionen, Rotationen etc.).

Zum Thema Interaktionsgrad: Wenn man eine Skala definieren würde, die von 100 Prozent Interaktion (zum Beispiel Erstellen einer Grafik per Lichtgriffel oder Grafik-Tablett) bis 0 Prozent (Einladen einer gespeicherten Grafik vom Massenspeicher) reichte, dann könnte man zum Beispiel Hi-Eddi zwischen 98 und 60 Prozent einordnen. Hires-3 bietet als Befehlsvorrat — je nach Einsatz im Programm — nahezu die gesamte Skala. Allerdings müßte man den Schwerpunkt der Interaktionsfähigkeit unterhalb von Hi-Eddi definieren, denn die Eingabemöglichkeiten per Joystick oder Lichtgriffel sind nicht besonders unterstützt, müssen also gesondert programmiert werden.

5. Systematik-Kauderwelsch

Kommen wir nun wieder zum Versuch zurück, grafische Datenverarbeitung zu strukturieren. Sinnvolle Unterteilungen der Grafik im engeren Sinn (Daten werden zu Bildern) hat kein Autor zu treffen versucht. Zu neu und in rasanter Entwicklung scheint dieses Gebiet zu sein.

Ich möchte deshalb nur auf zwei unvollständige Strukturen eingehen. Zum einen unterteilt man gerne nach dem Kriterium der Anwendung:

a) Business-Grafik

Dazu zählen zum Beispiel die allseits bekannten Balken-, Torten- und Liniendiagramme

b) CAD = »Computer aided design«

Darunter versteht man das Konstruieren mit Computerhilfe.

c) Ein Sammelsurium weiterer Anwendungsbereiche wie:

— Mathematische: Untersuchung von Funktionen, Lösungsmengen etc.

— Statistische etc.: Operations Research, Meßwertverteilung, etc.

— Künstlerische: Filmdesign, ...

— etc.

Aus der engen Verknüpfung von Grafik mit der Mathematik rührt eine andere Unterteilung her, die die Algorithmen und ihre Herkunft aus den mathematischen Teilgebieten als Kriterien hat:

a) 2-dimensionale Grafik (2D)

Analytische Geometrie der Ebene, ebene Transformationen, Approximationen, Schraffuren, etc.

b) 3-dimensionale Grafik (3D)

Analytische Geometrie des Raumes, räumliche Transformationen, Projektionen, hidden-line-Problem, etc.

Sie sehen sicher selbst, daß es äußerst schwierig scheint, hier in dieses Kauderwelsch eine gewisse Ordnung zu bringen. Wir werden in den weiteren Grafik-Folgen außer einigen grundlegenden programmtechnischen Details (wie zum Beispiel das »smooth scrolling«) Beispiele aus fast allen Gebieten erarbeiten.

Fortsetzung folgt

(Heimo Ponnath/gk)

Literatur:
[1] Theo Pavlidis: Algorithms for Graphics and Image Processing, Berlin-Heidelberg 1982: Springer-Verlag, ISBN 3-540-11338-X

Assembler ist keine Alchimie — Teil 12

Wie macht man Programme Reset-fest? Außerdem schauen wir dem »Rasterzeilen-Interrupt« etwas genauer auf die Finger und entwickeln ein Programm dazu.

Die Sache mit dem Modulstart

Sowohl beim RESET als auch beim NMI haben wir festgestellt, daß der Modulstart-Bereich ab $8000 eine besondere Rolle spielt. In Bild 5 finden Sie nochmal zusammengefaßt, was sich dort findet wenn ein Modul vorhanden ist.

Wir wollen im folgenden Beispielprogramm (Listing 1) ein Modul simulieren, indem wir den SMON mittels des RESET anspringen. Der NMI — also die RUN/STOP-RESTORE-Tastenkombination — soll dabei wirkungslos gemacht werden.

Bild 6 zeigt ein Flußdiagramm dieses Beispielprogrammes:

Achten Sie bitte darauf, daß Sie nach dem Eintippen des Programmes abspeichern und — natürlich — daß die SMON-Version ab $C000 im Speicher vorliegt. Mit SYS 24576 starten Sie unser Programm, in dem durch diesen SYS-Befehl zunächst nach $8000/1 die Startadresse einer neuen RESET-Serviceroutine geschrieben wird und nach $8002/3 die der neuen NMI-Routine. Außerdem wird die Modulkennung in die vorgeschriebenen Speicherplätze eingetragen. Wenn Sie nun mal die RESTORE-Taste — oder RUN/STOP und RESTORE — drücken, passiert offensichtlich nichts. Das liegt daran, daß unser Programm lediglich die auf den Stapel gelegten Register wieder zurückholt und aus der Unterbrechung mit RTI ins normale Geschehen zurückkehrt.

Haben Sie einen RESET-Taster eingebaut? Dann drücken Sie doch mal drauf. Zunächst erkennen Sie den normalen RESET-Verlauf. Dann meldet sich aber nicht wie gewohnt die Nachricht CBM-Basic..., sondern der SMON mit einer Registeranzeige. Das RESET-Programm ab $602E folgt dem Firmware-Programm. Lediglich der letzte Sprungbefehl ist anders und führt statt ins Basic in den SMON. Der SMON wird fehlerfrei funktionieren... solange Sie nicht versuchen, mit dem X-Kommando wieder ins Basic zurückzukehren. Dann wird Unsinn passieren, denn auf einen Start mittels RESET ist der SMON nicht gefaßt gewesen und in den Speicherstellen, die sonst eine Rückkehradresse enthalten, befindet sich nichts Sinnvolles. Es ist daher auch nicht möglich, den SMON wieder zu verlassen — außer durch Speicherstellenmanipulationen oder die Notbremse: Aus- und wieder Einschalten. Auf diese Weise (und mittels eines AUTOSTART) sichern sich Softwarehäuser manchmal gegen unbefugtes Kopieren ihrer Programme.

Nutzung der Unterbrechungen

Sowohl was die Hardware als auch die Firmware für die Unterbrechungsbehandlung angeht, haben wir nun einen guten Überblick gewonnen. Es ist jetzt an der Zeit, daß wir uns ansehen, auf welche Weise man dieses Reservoir an vielfältigen Möglichkeiten für sich nutzen kann. Dazu soll uns ein Überblick dienen:

I) Auslösung der Unterbrechung durch Hardware-Einwirkungen.

Da hätten wir beispielsweise den Userport oder den Expansion-Port, über die wir per CIAs Unterbrechungen anfordern können. Um es gleich zu sagen: Damit werden wir uns nicht auseinandersetzen. Meine Kenntnisse auf diesem Gebiet sind zu dünn. Aber vielleicht verstehen Sie das auch mal als Aufforderung, Ihre Versuche dazu anderen zu offenbaren? Also: Schreiben Sie doch mal!

II) Unterbrechungsauslösung per Software:

Damit haben wir immer noch ein weites Feld von Möglichkeiten vor uns:

IIa) Vorgesehene Nutzungen des IRQ

— mittels des VIC-II-Chips.

Da können wir uns auf den Rasterzeileninterrupt, die Sprite/Hintergrund- oder die Sprite/Sprite-Kollision stützen.

— oder mit Hilfe des CIA1

Da ist es vor allem der 60mal pro Sekunde auftretende Timer A-Unterlauf, der uns interessieren soll.

IIb) Vorgesehene Nutzungen des NMI

— CIA2: Läßt man die RS232C-Schnittstellenbehandlung außer acht, dann gibt es keine vorgesehene Nutzung.

— RESTORE: Zusammen mit der RUN/STOP-Taste kann man die vorgegebene Routine verändern, wie wir es schon in einigen Beispielen gezeigt haben.

Wir können außerdem noch unterscheiden zwischen Nutzungen, die periodisch stattfinden sollen (zum Beispiel eine spezielle Tastaturabfrage) und solchen, die stochastisch (= zufallsabhängig) oder willkürlich erfolgen (zum Beispiel Drücken der RESTORE-Taste). Beides ist auch durchführbar bei:

IIc) Nicht vorgesehene Nutzung der Unterbrechungen.

Da bietet sich vor allem der meistens völlig brach liegende CIA2 an mit seinen beiden Timern und der Alarmfunktion.

Wenn Sie aber erst einmal vertraut sind mit der Unterbrechungs-Programmierung und auch etwas Zeit zum Tüfteln investieren, finden Sie bestimmt noch eine ganze Menge weiterer Möglichkeiten.

Bei mehreren gleichartigen Unterbrechungsanforderungen (zum Beispiel IRQs) muß noch ein Weg gefunden werden, wie zwischen den dann vielleicht anfallenden unterschiedlichen Service-Routinen differenziert werden kann. Denkbar wären beispielsweise Aufgabenstellungen wie:

Jeder 3. Timer-IRQ soll den Joystick abfragen, oder RESTORE + h soll den Hilfsbildschirm zeigen, RESTORE + z soll den aktuellen Bildschirm wieder restaurieren, etc.

Sie sehen, eine große Menge Arbeit wartet auf uns. Nicht zu allen Möglichkeiten werde ich hier Beispielprogramme zeigen. Außerdem dürfen die dann auch nicht zu undurchsichtig sein und man sollte möglichst den Erfolg eines solchen Demo-Programmes auf dem Bildschirm erkennen können. Trotzdem hoffe ich, daß die nachfolgend und noch in der nächsten Folge gezeigten Programmlösungen ausreichen, Ihnen die Unterbrechungs-Behandlung mit eigenen Routinen durchschaubar zu machen. Ich will Ihnen aber nicht verschweigen, daß auch mir noch längst nicht alle Geheimnisse der Unterbrechungsprogrammierung offenbar geworden sind. Oft finde ich mich unversehens in Programm-Sackgassen wieder. Das soll Ihnen als kleiner Trost dienen, wenn Sie mal nach dem 1001. Absturz müde und mit rauchendem Kopf vor Ihrem Commodore-Ungeheuer sitzen.

Ein Programm zum VIC-II-IRQ

Sehr schöne Effekte lassen sich durch eine periodische IRQ-Anforderung per Rasterzeileninterrupt mittels des VIC-II-Chip erzielen. Deshalb ist sowas auch ein beliebtes Objekt für Demos von Unterbrechungsprogrammen. Als Ziel setzen wir uns, einen Bildschirm zu konstruieren, dessen Rahmen in allen Farben schillert.

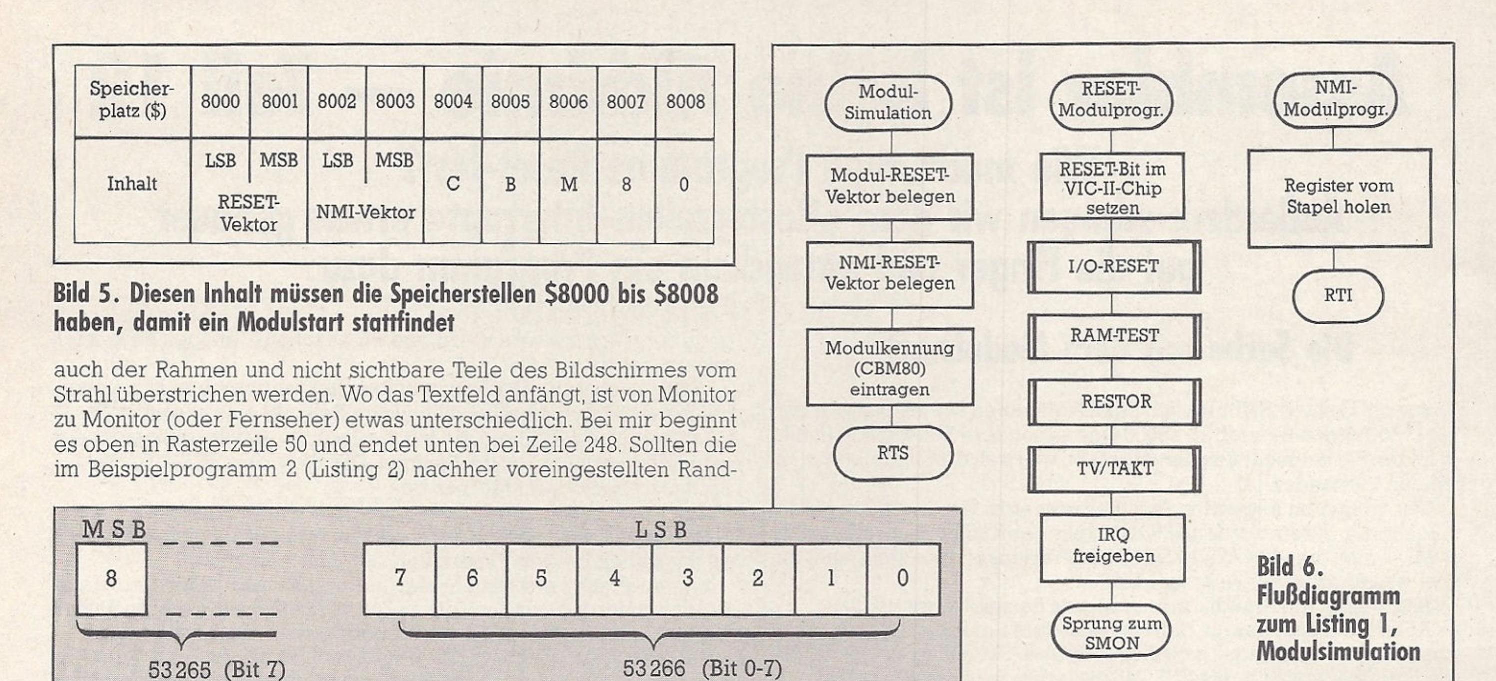
Leser der Grafikseite werden diese Möglichkeit des VIC-II-Chip schon kennen: Man kann dem Kathodenstrahl, der über den Monitor huscht, um das Bild zu erzeugen, über zwei Register folgen, die Rasterregister, wo jede Rasterzeile mitgezählt wird. Ohne an dieser Stelle allzusehr in die Einzelheiten einzugehen, soll hier nur bemerkt werden, daß die Numerierung dabei etwa von 0 bis 280 geht, weil

```
PROGRAMM 1

,6000  A9 2E     LDA #2E
,6002  8D 00 80  STA 8000
,6005  A9 60     LDA #60
,6007  8D 01 80  STA 8001
,600A  A9 41     LDA #41
,600C  8D 02 80  STA 8002
,600F  A9 60     LDA #60
,6011  8D 03 80  STA 8003
,6014  A9 C3     LDA #C3
,6016  A2 C2     LDX #C2
,6018  A0 CD     LDY #CD
,601A  8D 04 80  STA 8004
,601D  8E 05 80  STX 8005
,6020  8C 06 80  STY 8006
,6023  A9 38     LDA #38
,6025  A2 30     LDX #30
,6027  8D 07 80  STA 8007
,602A  8E 08 80  STX 8008
,602D  60        RTS
---------------------------
,602E  8E 16 D0  STX D016
,6031  20 A3 FD  JSR FDA3
,6034  20 50 FD  JSR FD50
,6037  20 8A FF  JSR FF8A
,603A  20 5B FF  JSR FF5B
,603D  58        CLI
,603E  4C 00 C0  JMP C000
---------------------------
,6041  68        PLA
,6042  A8        TAY
,6043  68        PLA
,6044  AA        TAX
,6045  68        PLA
,6046  40        RTI
```

Programm 1. Simulation eines Moduls





Speicherplatz ($)	8000	8001	8002	8003	8004	8005	8006	8007	8008
Inhalt	LSB	MSB	LSB	MSB	C	B	M	8	0
	RESET-Vektor		NMI-Vektor						

Bild 5. Diesen Inhalt müssen die Speicherstellen $8000 bis $8008 haben, damit ein Modulstart stattfindet

Bild 6. Flußdiagramm zum Listing 1, Modulsimulation

Bild 7. So sieht das 9-Bit-Register im VIC-II-Chip aus, welches die Rasterzeilen mitzählt

auch der Rahmen und nicht sichtbare Teile des Bildschirmes vom Strahl überstrichen werden. Wo das Textfeld anfängt, ist von Monitor zu Monitor (oder Fernseher) etwas unterschiedlich. Bei mir beginnt es oben in Rasterzeile 50 und endet unten bei Zeile 248. Sollten die im Beispielprogramm 2 (Listing 2) nachher voreingestellten Rand-

Bunter Rahmen Initialisierung
IRQ sperren
IRQ-Vektor neu belegen
Rasterzeile auf unteren Textfensterrand
Rasterzeilen-unterbrechung freigeben
— Rahmenfarbe schwarz
— Streifenbreite in Merkregister
IRQ freigeben
RTS

neuer IRQ
Prüfen, ob VIC-IRQ
VIC-IRQ ?
N
CIA1-IRQ-Register löschen
IRQ freigeben
Sprung zur normalen IRQ-Routine
J
Vergleich, ob Rasterzeile = unteres Textfenster
= unteres Textfenster ?
J
Rahmenfarbe schwarz
Rasterzeile auf oberen Textfensterrand setzen
Sprung zum Rest der normalen IRQ-Routine
N
Streifenbreite zum Rasterregister addieren
Verzögerungs-schleife
Rahmenfarbe +1
Sprung z. Rest d. normalen IRQ-Routine

Ausschalten der eigenen IRQ-Routine
IRQ-sperren
Rasterzeilen-Unterbrechung sperren
Normalen IRQ-Vektor wiederherstellen
Hellblau in Rahmenfarbreg.
IRQ freigeben
RTS

Bild 8. Flußdiagramm zum im Text vorgestellten Listing 2. Ein bunter Rahmen wird erzeugt.

werte bei Ihnen also anders sein, können Sie sie durch einige später noch angegebenen POKEs ändern. Die beiden Rasterzeilenregister sind:

$D012 (53266)
$D011 (53265)

Von $D011 allerdings ist nur das Bit 7 als msb der Rasterzeilenzahl für uns von Bedeutung. Bild 7 soll diese Belegung deutlich machen:

Das Interessante an diesen Registern ist nun, daß man auch in sie schreiben kann. Die auf diese Weise festgelegte Rasterzeile ist dann der Auslöser des IRQ, falls dieser im Interrupt-enable-Register $D01A freigegeben wurde (das kennen wir noch aus der letzten Folge).

Damit kann also unsere primäre Unterbrechungsquelle (der VIC-II-Chip) programmiert werden. Halten wir die zwei Schritte dazu nochmal fest:

1) Rasterzeile festlegen, bei der ein IRQ ausgelöst werden soll, durch Einschreiben in die Register $D012 und Bit 7 von $D011.
2) Freigeben des Rasterzeileninterrupts durch Einschreiben von 1000 0001 in das Interrupt-enable-Register $D01A.

Der nächste Schritt betrifft die Bearbeitung des IRQ durch die CPU. Wie wir vorhin sahen, springt das Programm beim IRQ mittels eines indirekten Sprunges, der auf den Vektor 788/9 ($314/5) zugreift. Dieser Vektor muß nun auf die eigene Routine verbogen werden, also:

3) Vektor $314/5 auf die IRQ-Service-Routine richten.

Damit wären alle Vorbereitungen getroffen. Der Rest liegt nun ganz bei uns — beziehungsweise bei dem von uns zu schreibenden Service-Programm. Als Bild 8 finden Sie ein Flußdiagramm unseres Beispielprogrammes 2.

Gehen wir nun an die Realisierung. Zunächst also die Initialisierung, die wir bei $6000 (also durch SYS 24576 zu starten) beginnen lassen:

6000	**SEI**	Sperren von IRQs
Schritt 3:		
6001	**LDA #$28**	LSB der IRQ-Routine
6003	**STA 0314**	in IRQ-Vektor-LSB
6006	**LDA #$60**	MSB der IRQ-Routine
6008	**STA 0315**	in IRQ-Vektor-MSB
Schritt 1:		
600B	**LDA #$F8**	Rasterzeile, bei der das Textfenster endet. Von da an soll der Rahmen schwarz sein.
600D	**STA D012**	in Rasterzeilen-Register (LSB) schreiben.
6010	**LDA D011**	Register mit dem msb des Rasterzeilenzählers
6013	**AND #$7F**	0111 1111 löscht das Bit7
6015	**STA D011**	Zurückschreiben. Damit ist die Rasterzeile, die den IRQ auslösen soll, festgelegt.
Schritt 2:		
6018	**LDA #$81**	1000 0001 wird nun
601A	**STA D01A**	ins IRQ-enable-Register geschrieben, um den Rasterzeilen-IRQ zuzulassen.

Festlegen einiger Startwerte:

601D	**LDA #$00**	Farbe schwarz
601F	**STA D020**	in Rahmen schreiben
6022	**LDA #$04**	Streifenbreite in
6024	**STA 02**	Merkregister schreiben.
6026	**CLI**	IRQ freigeben
6027	**RTS**	Ende der Initialisierung.

Von nun an laufen alle IRQs über unsere eigene Routine, die bei $6028 beginnt.

Zunächst müssen wir prüfen, ob die Unterbrechung vom VIC-II-Chip kommt oder vom CIA1:

6028	**LDA D019**	IRQ-Request-Register des VIC-II-Chip (siehe letzte Folge). Dort ist Bit 7 gesetzt, wenn die Anforderung vom VIC-II-Chip kam.
602B	**STA D019**	Zurückschreiben
602E	**BMI 6037**	Sprung, falls VIC-IRQ, sonst CIA-IRQ.
Bearbeiten eines CIA-IRQ:		
6030	**LDA DC0D**	Löschen des CIA1 Unterbrechungs-Kontrollregisters.
6033	**CLI**	IRQ zulassen. Damit können innerhalb eines CIA- IRQ auch unsere VIC-IRQs geschehen.
6034	**JMP EA31**	Bearbeitung des CIA-IRQ durch die normale Routine.

Unser Programm für VIC-II-IRQs:

6037	**LDA D012**	Rasterzeilen-Register laden um festzustellen, welche Zeile den IRQ auslöste.
603A	**CMP #$F8**	Vergleich mit Ende des Textfensters.
603C	**BCS 604F**	Wenn unterhalb des Textfensters, Sprung.

Der folgende Programmteil ist wirksam, wenn der IRQ-Auslöser eine Zeile in Höhe des Textfensters war:

603E	**CLC**	Addition vorbereiten.
603F	**ADC 02**	Streifenbreite aus dem Merkregister addieren.
6041	**STA D012**	Neuen Wert in Rasterzeilen-Register schreiben.

Damit wird eine neue Rasterzeile als IRQ-Auslöser festgelegt, die um die Streifenbreite tiefer liegt als die vorhergegangene.

Es folgt eine kleine Verzögerungsschleife, die aber nur zum Experimentieren eingebaut wurde:

6044	**LDY #$03**	Schleifen-Startwert
6046	**DEY**	Herunterzählen
6047	**BNE 6046**	NEXT Y, bis Y=0.
Ändern der Rahmenfarbe bis zum nächsten Raster-IRQ:		
6049	**INC D020**	Farbcode+1. Wenn Code im Rahmenfarbregister größer als 15 wird, fängt wieder Farbcode 0 an, weil die Bits 5-7 keine Funktion haben.

Abschließend erfolgt der Rücksprung in den Rest der normalen IRQ-Routine:

604C	**JMP EA81**	Siehe unsere Untersuchung der IRQ-Firmware.

Damit ist der Rahmen in Höhe des Textfensters behandelt. Es schließt sich nun der Teil an, der die Rahmenbereiche unter- und oberhalb bearbeitet:

604F	**LDA #$00**	Farbcode schwarz
6051	**STA D020**	in Rahmenfarb-Register.
6054	**LDA #$32**	Rasterzeile, bei der oben das Textfenster beginnt.
6056	**STA D012**	In Rasterzeilen-Register schreiben
6059	**JMP EA81**	Abschluß durch Sprung zum Ende der normalen IRQ- Routine.

Damit ist festgelegt, daß ober- und unterhalb des Textfensters die Rahmenfarbe schwarz wird.

Unsere eigene Routine ist jetzt abgeschlossen. Zum guten Ton gehört es, dem Benutzer auch die Möglichkeit zu öffnen, diese Routine wieder abzuschalten. Das erfolgt im letzten Programmteil, der mittels SYS24688 aktiviert werden kann:

605C	**SEI**	IRQ sperren
605D	**LDA #$00**	Raster-IRQ
605F	**STA D01A**	abschalten
6062	**LDA #$31**	IRQ-Vektor
6064	**STA 0314**	restaurieren

Fortsetzung auf Seite 131

```
PROGRAMM 2

,6000  78        SEI
,6001  A9 28     LDA #28
,6003  8D 14 03  STA 0314
,6006  A9 60     LDA #60
,6008  8D 15 03  STA 0315
,600B  A9 F8     LDA #F8
,600D  8D 12 D0  STA D012
,6010  AD 11 D0  LDA D011
,6013  29 7F     AND #7F
,6015  8D 11 D0  STA D011
,6018  A9 81     LDA #81
,601A  8D 1A D0  STA D01A
,601D  A9 00     LDA #00
,601F  8D 20 D0  STA D020
,6022  A9 04     LDA #04
,6024  85 02     STA   02
,6026  58        CLI
,6027  60        RTS
--------------------------
,6028  AD 19 D0  LDA D019
,602B  8D 19 D0  STA D019
,602E  30 07     BMI 6037
,6030  AD 0D DC  LDA DC0D
,6033  58        CLI
,6034  4C 31 EA  JMP EA31
--------------------------
,6037  AD 12 D0  LDA D012
,603A  C9 F8     CMP #F8
,603C  B0 11     BCS 604F
,603E  18        CLC
,603F  65 02     ADC   02
,6041  8D 12 D0  STA D012
,6044  A0 03     LDY #03
,6046  88        DEY
,6047  D0 FD     BNE 6046
,6049  EE 20 D0  INC D020
,604C  4C 81 EA  JMP EA81
--------------------------
,604F  A9 00     LDA #00
,6051  8D 20 D0  STA D020
,6054  A9 32     LDA #32
,6056  8D 12 D0  STA D012
,6059  4C 81 EA  JMP EA81
--------------------------
,605C  78        SEI
,605D  A9 00     LDA #00
,605F  8D 1A D0  STA D01A
,6062  A9 31     LDA #31
,6064  8D 14 03  STA 0314
,6067  A9 EA     LDA #EA
,6069  8D 15 03  STA 0315
,606C  A9 0E     LDA #0E
,606E  8D 20 D0  STA D020
,6071  58        CLI
,6072  60        RTS
--------------------------
```

Listing 2. Das im Artikel entwickelte Programm auf einen Blick

Memory Map mit Wandervorschlägen (10)

Diesmal geht es weiter mit den Speicherzellen 158 bis 182. Behandelt werden dabei die interne Uhr, Kassettenoperationen und die RS232-Schnittstelle.

In der Ausgabe 6/85 habe ich unter anderem auch die Speicherzelle 144 behandelt. In einem separaten Texteinschub »ST-ATUS« bin ich näher auf die Status-Variable ST eingegangen.

Ganz am Schluß, nach der Besprechung der Floppy-Operationen, habe ich noch den Wert ST=128 (DEVICE NOT PRESENT) behandelt.

Ich mußte allerdings eingestehen, daß mir nicht bekannt war, wie man den Wert 128 innerhalb eines Programms abfragen könnte, um eine Programmunterbrechung bei abgeschalteten Peripherie-Geräten zu vermeiden.

Dieses Eingeständnis hatte unerwartete Folgen. Ich habe mehrere Zuschriften erhalten, die mir Hinweise gaben, wie das Problem zu lösen ist:
— Herr Witte aus Wunstorf
— Herr Bojko aus Dortmund
— Herr Stahl aus Stuttgart
— Herr Kühlewind aus Berchtesgaden
— Herr Möller und Dipl.-Ing. Minning aus Marburg
— Herr Müller aus Wesel

Ich muß ehrlich sagen, daß ich mich über soviel »Mitarbeit« sehr gefreut habe. Allen Schreibern danke ich hiermit herzlich für die Hinweise.

Ich kann hier nicht alle Lösungen vorstellen. Es ist auch nicht notwendig, da sie alle sehr ähnlich sind. Ich habe deswegen eine Zusammenfassung erstellt, die — wie ich hoffe — allen Beiträgen gerecht wird.

Es gibt zwei Speicherzellen — 768/769 —, auf die wir bei unserer Wanderung durch die Speicherlandschaft noch kommen werden, in denen in Low-/High-Byte-Darstellung eine Adresse steht, auf die das Betriebssystem springt, wenn die Meldung »DEVICE NOT PRESENT« ausgegeben werden soll. Diesen Zeiger kann man so »verbiegen«, daß die Meldung nicht ausgegeben wird, und daß das Programm einfach weiterläuft.

Normalerweise steht in 768 die Zahl 139 (VC 20: 58), in 769 die Zahl 227.

Verbogen wird der Zeiger durch eine 61 (185 geht auch), beim VC 20 durch 52. Dadurch zeigt die Adresse auf eine Speicherzelle des Betriebssystems, in welcher der Assembler-Befehl »RTS«, das bedeutet Rücksprung, steht. Jetzt können wir ungestört den STatus abfragen, wir müssen allerdings den negativen Wert von ST, also —128 nehmen.

Für die Floppy sieht die Abfrage fast gleich aus.

Die einzige Änderung ist in den Zeilen 20 und 30:
20 OPEN 1,8,15
30 diese Zeile entfällt.

Unter bestimmten Umständen kann das Verbiegen des Zeigers in 768 entfallen, wie einige Zuschriften ergeben haben.

Ich möchte nach eigenen längeren Versuchen aber dafür plädieren, die Fehlermeldung immer abzuschalten, um nie in Schwierigkeiten zu kommen.

Vorsicht ist die Mutter der Weisheit.

Für den Drucker sieht das so aus:

```
 10 POKE 768,61                  Fehlermeldung abschalten
 20 OPEN 1,4                     Gerät ansprechen
 30 PRINT #1
 40 CLOSE 1
 50 POKE 768,139                 Fehlermeldung einschalten
 60 IF ST=—128 THEN 100          Sprung bei ausgeschaltetem
                                 Gerät
 70 PRINT" FORTSETZUNG"          Weiter im Programm
 80 END                          Ende der DEMO
100 PRINT" GERAET EINSCHALTEN"
110 GET A$:IF A$=""THEN 110      Warteschleife
120 GOTO 10                      neuer Versuch
```

Adresse 158 und 159 ($9E und $9F)

Zwischenspeicher bei Kassettenoperationen

Diese beiden Speicherzellen werden von Routinen des Betriebssystems verwendet, welche bei Kassettenoperationen die Zeichen überprüfen, ob sie richtig sind, und welche bei aufgetretenen Fehlern Korrekturen durchführen.

Adresse 160 bis 162 ($A0 bis $A2)

Interne Uhr für TI und TI$

Das Basic der Commodore-Computer kennt neben der Variablen ST (siehe Speicherzelle 144) noch zwei weitere »reservierte« Variable, nämlich TI und TI$. Beide bieten eine interne Uhr, welche aus dem Inhalt der Speicherzellen 160 bis 162 abgeleitet wird. Diese drei Zellen funktionieren wie der Kilometerzähler eines Autos, halt nur mit drei Stellen.

Die hinterste Stelle ist die Zelle 162. Ihr Inhalt wird beim Einschalten des Computers auf 0 gesetzt, dann aber 60mal in der Sekunde um 1 erhöht. Das erfolgt durch die automatische Interrupt-Routine, welche auch die STOP-Taste abfragt und noch andere Hausaufgaben 60mal in der Sekunde ausführt. Da 1/60 = 0,01667 ist, zählt also die Zelle 162 in 0,01667 Sekunden um 1 weiter. Sie kann wie alle Speicherzellen maximal nur die Zahl 255 enthalten, danach kommt wieder eine 0. Das heißt aber, daß sie nach 256 * 0,01667 = 4,267 Sekunden einmal durchgelaufen ist.

Nach jedem Durchlauf wird die davorliegende Speicherzelle 161 um 1 erhöht. Sie zählt also in 4,267 Sekunden um 1 weiter und ist nach 256 * 4,067 = 1 092,26 Sekunden oder besser nach 18,2044 Minuten einmal durchgelaufen. Nach dem Kilometerzähler-Prinzip wird nach jedem Durchlauf von 161 der Inhalt der davorliegenden Zelle 160 um 1 erhöht.

Die Zelle 160 zählt also in 18,2044 Minuten um 1 weiter und ist nach 256 * 18, 2044 = 4660,34 Minuten, das sind 77,67 Stunden, einmal durchgelaufen.

Diese Stundenzahl wird allerdings niemals erreicht, da das Betriebssystem nach Erreichen des Wertes für 24 Stunden alle drei Zellen wieder auf 0 zurücksetzt. Wir werden das gleich nachprüfen.

Zuerst aber wollen wir uns den dreizelligen Zähler anschauen:
10 PRINT PEEK(160);PEEK(161);PEEK(162)
20 GOTO 10

Nach RUN sehen wir den Inhalt der drei Zellen sich entsprechend der oben angegebenen Zeiten verändern. Die Zahlen sind nicht vorherbestimmbar, denn der Zähler ist ja nach dem Einschalten des Computers schon losgelaufen. Er kann aber auf 0 gesetzt werden durch Einfügen der Zeile 5:
5 POKE 160,0:POKE 161,0:POKE 162,0

Jetzt beginnt der Zähler immer ab 0. Ich habe gerade gesagt, daß der Zähler auf 0 gesetzt wird, wenn er 24 Stunden lang gelaufen ist. Der Inhalt in den drei Speicherzellen, der 24 Stunden entspricht, ist nach der oben angegebenen Umrechnungsart 79-26-0. Diesen Wert, oder besser noch ein Wert kurz davor, in die Zellen 160 bis 162 gePOKEt, zeigt uns den Nullsetzvorgang. Ersetzen Sie bitte die obige Zeile 5 durch eine neue Zeile:
5 POKE 160,79:POKE 161,25:POKE 162,0

Nach vier Sekunden Laufzeit schalten alle drei Zellen in der Tat auf 0 zurück.

Die Umsetzung der Zahlen aus 160 bis 162 in die Variablen TI und TI$ sowie deren Wirkungsweise, entnehmen Sie bitte dem nebenstehenden Texteinschub 1 »Die eingebaute Uhr«.

Abschließend muß eines noch warnend erwähnt werden. Alle Operationen, welche den Interrupt-Vektor verwenden beziehungsweise verändern, stören oder verzögern die normale Interrupt-Routine, die ja den Zähler weiterstellt. So zählt der Zähler nicht gleichmäßig und die daraus abgeleitete Uhr geht nicht mehr richtig. Ein Beispiel dafür sind alle Ein- und Ausgaben über die Datasette, welche über einen Interrupt laufen.

Adresse 163 bis 164 ($A3 und $A4)

Zwischenspeicher

Diese beiden Speicherzellen werden von den Ein- und Ausgabe-Routinen des Betriebssystems für Kassetten, Floppy-Laufwerk und Drucker als Zwischenspeicher für alle möglichen Werte benutzt.

Adresse 165 ($A5)

Bit-Zähler für Kassetten-Synchronisierung

Beim Abspeichern eines Programms auf ein Band werden vor den eigentlichen Daten mehrere Bits zusätzlich gespeichert, die beim Einlesen dieses Bandes zur Synchronisierung dienen, das heißt zum Übereinstimmen der Geschwindigkeit der Datenübertragung.

Die Speicherzelle 165 wird als Zähler dieses Synchron-Bits verwendet.

Adresse 166 ($A6)

Zähler der bearbeiteten Bytes im Kassetten-Puffer

Diese Speicherzelle wird als Zähler benutzt, welcher angibt, wieviele Bytes gerade in den Kassetten-Puffer eingeschrieben oder aus ihm ausgelesen worden sind. Der Kassetten-Puffer besteht aus den Speicherzellen 828 bis 1 019 und kann somit 191 Byte aufnehmen, was zugleich die höchste Zahl ist, welche sinnvollerweise in der Zelle 166 stehen kann.

Nähere Erklärungen und ein paar Experimente mit Zelle 166 finden Sie in dem nebenstehenden Texteinschub 2 »Experimente mit dem Kassetten-Puffer«.

Die meisten der nächsten 20 Speicherzellen werden bei Operationen mit der RS232-Schnittstelle, die über den User-Port den Computer mit anderen Geräten verbindet, eingesetzt. Da die Programmierung der RS232-Schnittstelle nicht zu den üblichen Programmierarbeiten zählt, sondern nur von wenigen Spezialisten verwendet wird, gehe ich auf diese RS232-Adressen nicht im Detail ein. Dies sollte einem eigenen Spezialkurs vorbehalten sein.

Adresse 167 ($A7)

Zwischenspeicher für Kassetten-Operationen und für Eingabe über die RS232-Schnittstelle

Diese Speicherzelle wird verwendet, um jedes Bit, welches von einem RS232-Kanal über den User-Port eingelesen wird, zwischenzuspeichern.

Außerdem verwenden mehrere Kassetten-Routinen diese Adresse als Zwischenspeicher.

Adresse 168 ($A8)

Bitzähler für RS232-Eingabe und bei Band-Ein-/Ausgabe

Die Speicherzelle 168 wird als Zähler verwendet, der diesmal nicht die Bytes, sondern die Anzahl der Bits zählt, die sowohl über den User-Port als auch über den Kassetten-Port geleitet werden. Das dient dem Betriebssystem dazu, zu wissen, wann ein volles Wort abgearbeitet worden ist.

Adresse 169 ($A9)

RS232-Flagge für Startbit-Prüfung

Ein RS232-Datentransfer prüft, ob ein Start-Bit empfangen worden ist. Im positiven Fall steht in Zelle 169 die Zahl 144, im negativen Fall eine 0.

Adresse 170 ($AA)

RS232-Eingabe- und Zwischenspeicher für Kassetten-Routinen

Bei der Speicherzelle 165 haben wir gesehen, daß ein Band Synchronisationsbits enthält. Die Speicherzelle 170 wird dabei als Flagge benutzt, die angibt, ob ein gelesenes Zeichen Synchronisierungs-Bits oder ein Datenwort darstellt.

Die RS232-Routinen verwenden Zelle 170 dagegen als Speicher, in welchem die eingelesenen Bits zu einem Byte zusammengefaßt werden, bevor sie im Eingabepuffer am oberen Ende des Programmspeichers abgelegt werden (siehe auch Speicherzellen 55/56).

Adresse 171 ($AB)

Quersummenprüfung und Zähler für Band-Header bei RS232- und Kassetten-Operationen

Diese Speicherzelle wird vom Betriebssystem benutzt um festzustellen, ob während einer RS232-Datenübertragung Bits verloren wurden. Da derartige Prüfungen mit Parity-Bits (Quersummenprüfung) des öfteren erwähnt werden, gebe ich eine kurze Beschreibung des Prüfprinzips im nebenstehenden Texteinschub 3 »Fehlererkennung mit Parity-Bits«.

Zusätzlich wird in 171 die Länge des Band-Vorspanns bei seiner Erzeugung gezählt.

Adresse 172 und 173 ($AC und $AD)

Zeiger auf die Anfangsadresse für Ein-/Ausgabe, Zwischenspeicher für den Bildschirmeditor

In den Speicherzellen 193/194 steht ein Zeiger, der auf die Adresse im Programmspeicher zeigt, wo das Programm beginnt beziehungsweise beginnen soll, welches abgespeichert beziehungsweise geladen werden soll.

Dieser Zeiger wird am Anfang einer Lade- oder Abspeicher-Operation in die Zellen 172/173 gebracht, wo er während der Operation laufend erhöht wird, bis das Ende des Programms erreicht ist; dann wird er wieder auf seinen ursprünglichen Wert gesetzt.

Der Zeiger dient außerdem noch dem Bildschirmeditor als Zwischenspeicher während des Scrollens (Hochschieben) des Bildschirms und beim Einfügen zusätzlicher Zeilen.

Dieser Zeiger kann sehr nützlich sein, um Programme entweder schon beim SAVEn oder aber erst beim LOADen gezielt auf andere als ursprüglich verwendete Speicherbereiche zu bringen. Dazu sind aber noch einige andere Zellen notwendig, bis hin zu dem schon erwähnten Zeiger in 193/194. Ich werde mit dieser Anwendung und ihrer Beschreibung daher warten, bis wir zu 193/194 kommen.

Adresse 174 und 175 ($AE und $AF)

Zeiger auf die Endadresse für Ein-/Ausgabe, Zwischenspeicher für den Bildschirmeditor

Dieser Zeiger ist der Zwilling zu 172/173, nur zeigt er seinerseits auf die letzte Adresse des zu bewegenden Programms (siehe oben).

Adresse 176 und 177 ($B0 und $B1)

Zeitkonstante

Der Wert in dieser Speicherzelle wird verwendet, um die Zeitkonstante zum Lesen vom Band in der Zelle 146 einzustellen.

Adresse 178 und 179 ($B2 und $B3)

Zeiger auf den Kassetten-Puffer

Beim Einschalten des Computers werden diese Speicherzellen in Low-/High-Byte-Darstellung auf die Anfangsadresse des Kassetten-Puffers gesetzt. Beim VC 20 und C 64 ist dies die Adresse 828 ($33C).

Adresse 180 ($B4)

RS232-Bit-Zähler und -Zwischenspeicher für Kassetten-Operationen

Die RS232-Routinen verwenden die Speicherzelle 180, um die Zahl der übertragenen Bits zu zählen, außerdem für Parity-Berechnung (siehe Texteinschub 3) und Stop-Bit-Bearbeitung. Die Lade-Routinen für Kassettenbetrieb benutzen diese Zelle als Flagge, die angibt, ob der Computer bereit ist, Daten zu übernehmen.

Adresse 181 ($B5)

RS232-Anzeige für nächstes Bit, Flagge für End-of-Tape

Bei RS232-Operationen enthält die Zelle 181 das jeweils nächste Bit, welches übertragen werden soll. Bandoperationen entnehmen dieser Speicherzelle, welcher Block gerade gelesen wird.

Adresse 182 ($B6)

Ausgabe-Zwischenspeicher für RS232 und Kassette

Bei Ausgabe von Daten über die RS232-Schnittstelle wird jedes Byte in seine Einzelteile zerlegt, bevor es über den Ausgabepuffer seriell übertragen wird. Der Ausgabepuffer wird im obersten Teil des Programmspeichers angelegt (siehe auch Speicherzellen 55 und 56); die genaue Anfangsadresse steht in Speicherzelle 248. Auch die Ausgabe von Daten auf die Kassette verwendet Zelle 182 als Ausgabe-Zwischenspeicher.

(Dr. H. Hauck/ah)

Texteinschub # 1 Die eingebaute Uhr

Der VC 20 und der C 64 haben eine interne Uhr eingebaut, deren Stand abgefragt, ausgedruckt und somit zur Zeitmessung und Programmsteuerung eingesetzt werden kann.

In der Basic-Befehlsliste der Handbücher finden wir dazu zwei Funktionen, TI und TI$.

1) TI gibt den Stand des Zählers wieder, der durch die drei Speicherzellen 160, 161 und 162 gebildet wird. Dabei ist der Wert von TI nichts anderes als die Summierung des Inhalts dieser drei Zähler.

Entsprechend dem dreistelligen Zählerprinzp (siehe Beschreibung der Speicherzellen 160 bis 162) ist die Summe:

```
TI  = Inhalt (162)
    + Inhalt (161) * 256
    + Inhalt (160) * 256 * 256
```

Mit dem folgenden kleinen Programm können wir das verifizieren:

```
10  PRINT TI;
20  PRINT PEEK(162)+256*PEEK(161)+256*256*PEEK(160)
30  GOTO 10
```

Die beiden Zahlenbänder für TI und die Zählersumme sind praktisch identisch.

2) TI$ gibt ebenfalls den Stand des Zählers wieder, aber in einer anderen Darstellung. Während TI 60mal in der Sekunde weiterzählt, gibt TI$ direkt Stunden, Minuten und Sekunden an.

Den Zusammenhang zwischen TI und TI$ können Sie am besten mit dem folgenden kleinen Programm sehen:

```
10  PRINT INT(TI/60);
20  PRINT TI$
30  GOTO 10
```

Zeile 10 rechnet TI in Sekunden um. Damit die Zeile nicht mit vielen Dezimalstellen volläuft, verwandelt sie das Resultat in eine ganze Zahl. Zeile 20 zeigt dazu im Vergleich die sechs Ziffern von TI$.

Das erste, was beim Ablauf des Programms auffällt, ist die gleichzeitige Umschaltung beider Zahlenreihen. Die Umrechnung von TI$ nach TI geht am besten »zu Fuß«. Stoppen Sie den Lauf mit der STOP-Taste. Nehmen Sie dann den letzten Wert von

TI$ (rechts). Die ersten beiden Ziffern sind die Stunden, ihr Wert wird mit 3600 multipliziert, um sie in Sekunden umzurechnen. Addieren Sie dazu den Wert der mittleren beiden Ziffern (Minuten) multipliziert mit 60, und addieren Sie zu diesem Zwischenergebnis die Sekunden (Ziffern ganz rechts). Das Resultat ist identisch mit dem letzten Wert von TI.

Wenn Sie übrigens den Ausdruck für TI$ optisch verbessern wollen, dann setzen Sie zwischen die Stunden, Minuten und Sekunden einen Doppelpunkt. Das wird durch eine String-Manipulation erreicht:

```
Print LEFT$(TI$,2)":"MID$(TI$,3,2)":"RIGHT$(TI$,2)
```

Eine gute Uhr muß sich stellen lassen — bei TI$ erreichen wir das einfach mit Zuweisen des gewünschten Wertes an die Variable TI$. Zum Beispiel stellt

```
TI$ = "153000"
```

die Uhr auf 15 Uhr 30. Man kann sie dementsprechend auch auf 0 zurücksetzen, was bei einem Stoppuhr-Betrieb notwendig wird.

TI kann direkt nicht beeinflußt werden, nur über POKEn von neuen Werten in die Speicherzellen 160 bis 162 oder durch die Zuweisung von Werten an TI$.

Die eleganteste Methode, TI und TI$ auf 0 zu setzen, geht beim C 64 und VC 20 mit

```
SYS 65499
```

Wenn Sie noch das kleine Programm von oben im Rechner haben, können Sie es gleich ausprobieren. Geben Sie direkt ein:

```
SYS 65499:RUN
```

und die Uhr startet von Null an.

Abschließend möchte ich Ihnen noch zwei kleine Anwendungsbeispiele von TI und TI$ mitgeben. Das erste ist ein Kochrezept, wie die Laufzeit eines Programms gemessen werden kann. Diese Programm-Stoppuhr besteht aus zwei Zeilen.

Die erste Zeile setzt die Uhr auf 0, das kennen wir schon.

Die zweite Zeile druckt am Ende des Programms die abgelaufene Zeit aus.

```
10 TI$ = "000000"
:
:
10000 PRINT TI/60 "SEKUNDEN"
```

Das zu messende Programm steht zwischen diesen beiden Zeilen.

Das zweite Beispiel betrifft eine Uhr, die nach einer vorgegebenen Zeit ein Programm (Spiel) abbricht. Davon zeige ich zwei Versionen. Die eine Version ist nach allen Erklärungen von oben beinahe trivial:

```
10 TI$ = "000000"
1000 IF TI$ >"000700" THEN STOP
```

Diese beiden Zeilen setzen die Uhr auf 0 und brechen ein Programm nach genau 7 Minuten ab.

Etwas kniffliger ist der Abbruch (oder Start) mit einer Count-Down Uhr.

```
10   TI$ = "000000"
20   ZEIT = 300
30   IF ZEIT—VAL(TI$) < =0 THEN STOP
40   weiteres Programm
```

Die Variable »Zeit« gibt die Dauer des Count-Down in Sekunden an. Zeile 30 überprüft den Wert von TI$, bis er 300 erreicht hat, indem sie den jeweiligen Wert von TI$ von der vorgegebenen Zeit subtrahiert. Natürlich müssen in beiden Versionen die Prüfzeilen sinnvoll in ein Programm eingebaut werden. Aber das möchte ich gern Ihnen überlassen.

Texteinschub # 2 Experimente mit dem Kassetten-Puffer

Die Speicherzellen von 828 bis 1019 werden als »Kassetten-Puffer« bezeichnet.

Beim Abspeichern auf eine Kassette wird zuerst der Vorspann eines Bandes, der sogenannte »Header«, in diesen Puffer gespeichert. Ein Programm wird dann direkt auf das Band geschrieben. Eine Datei allerdings läuft zuerst auch in den Kassetten-Puffer und von dort erst auf das Band. Sie kennen sicher die charakteristischen Wartezeiten des Kassettenmotors beim SAVEn einer Datei.

Beim Laden von einer Kassette gilt der Unterschied zwischen einem Programm und einer Datei genauso, einschließlich der Benutzung des Kassetten-Puffers.

Wir haben gelernt, daß in der Speicherzelle 166 die Zahl der Bytes gezählt wird, die in den Puffer geschrieben, beziehungsweise aus dem Puffer gelesen worden sind. Die Zahl reicht von 0 bis 191.

Diese Speicherzelle 166 kann während eines Programms abgefragt und auch mit POKE beliebig verändert werden. Was dabei herauskommt, ist vordergründig nur eine Spielerei. Aber vielleicht kann man die folgenden Experimente auch nutzbringend einsetzen.

Zuerst wollen wir die Funktionsweise von 166 erproben. Dazu laden wir eine simple Datei auf ein leeres Band, und zwar mit folgendem Programm:

Programm # 1

```
10   OPEN 1,1,1
20   FOR I=100 TO 150
30   PRINT #1,I
40   NEXT
50   CLOSE 1
```

Wir eröffnen eine Datei (ohne Namen) mit der Nummer 1, für Kassette (die zweite 1), zum Schreiben (die dritte 1). Nach RUN wird der Kassetten-Puffer mit den Zahlen 100 bis 150 in mehreren Schüben gefüllt, wobei jeder Schub einzeln auf das Band geschrieben wird.

Den Zusammenhang zwischen den Datei-Zahlen und dem Zähler in 166 zeigt uns das folgende Ausleseprogramm:

Programm # 2

```
10   OPEN 1,1,0
20   GET #1,X$
30   Print X$;
40   PRINT CHR$(28)PEEK(166)CHR$(154);
50   GOTO 20
```

Wir eröffnen wieder eine Datei, diesmal zum Lesen (die 0), und bringen mit GET # die einzelnen Zeichen hintereinander in den Puffer und dann auf den Bildschirm. Die Zeile 40 druckt nach jedem Zeichen in roter Farbe [CHR$(28)] den Zählerstand und schaltet dann mit CHR$(154) — beim VC 20 wäre das CHR$(31) — wieder auf die Normalfarbe zurück.

Zuerst muß das Band zurückgespult werden, und dann geht es los mit RUN. Nach dem Erscheinen der ersten Zeichen auf dem Bildschirm stoppen Sie bitte den Ablauf mit der STOP-Taste.

Sie sehen jetzt in Rot den Inhalt der Zelle 166, die aufwärts zählt, und dazwischen in Blau die Zahlen von 100 aufwärts. Interessant ist, daß durch Zwischenräume für eine 3stellige Zahl 6 Byte verbraucht werden.

Fahren Sie mit CONT solange fort, bis der Kassettenmotor anläuft und der nächste Schub auf dem Bildschirm ausgedruckt wird. Nach erneutem STOP sehen Sie, daß die roten Zahlen nach 190 wieder auf 0 zurückspringen. Das war der Moment, wo der Kassettenmotor wieder eingeschaltet wurde.

Diese Erkenntnis verwenden wir für ein Experiment.

Mit der in das Programm # 2 eingeschobenen Zeile 45 fragen wir den Inhalt von 166 ab und beeinflussen damit dem Ablauf des Programms. Außerdem setzen wir an dieser Abfragestelle den Inhalt der Zelle 166 auf den Endwert 191 und zwingen damit den Kassettenmotor weiterzulaufen.

```
45 IF PEEK(166) = 18 THEN POKE 166,191
```

Die Wiederholung des Programms mit zurückgespultem Band bringt uns ein neues Ergebnis:

Sobald der Zähler in 166 die 18 erreicht hat, glaubt das Programm, der Kassetten-Puffer wäre bereits ausgelesen, schaltet den Kassettenmotor wieder ein und liest den nächsten Zahlenblock in den Puffer. Wir erhalten jetzt nicht alle Zahlen, die auf dem Band stehen, sondern nur Gruppen von 18 Bytes, das sind ungefähr drei Zahlen.

Ich sage »ungefähr« mit Absicht, denn mit der Symmetrie, beziehungsweise mit der richtigen Reihenfolge klappt es nicht immer so ganz, da ja die Länge des Kassetten-Puffers nicht unbedingt ein ganzzahliges Vielfaches der ausgelesenen Bytes ist. Da liegt also ein kleines Problem.

Dieses Abfragen und Abändern der Speicherzelle 166 geht natürlich auch in der anderen Richtung, nämlich beim Abspeichern von Zahlen. Nehmen Sie bitte nochmal das Programm # 1 her und ändern Sie es wie folgt ab:

Programm # 1.a

```
10   OPEN 1,1,1
20   FOR I=100 TO 300
30   PRINT#1,1
35   IF PEEK(166)=18 THEN POKE 166,191
40   NEXT
50   CLOSE 1
```

Wir haben jetzt die Abfrage der Speicherzelle 166 des Programms # 2 von vorhin in das Programm # 1 eingebaut. Spulen Sie bitte das Band zurück und lassen Sie das Programm laufen.

Nun wollen wir die dadurch neu abgespeicherte Datei ganz normal auslesen. Dazu nehmen wir das Programm #2, also ohne die Zeile 45. Das sieht dann so aus:

Programm #2.a

```
10 OPEN 1,1,0
20 GET#1,X$
30 PRINT X$;
40 PRINT CHR$(28)PEEK(166)CHR$(154);
50 GOTO 20
```

Wir starten es mit RUN, nachdem das Band wieder zurückgespult ist. Der Vorgang ist im Prinzip der gleiche wie bei Programm #2; halten Sie das Programm bitte auch wieder an, so wie vorher.

Wir sehen aber einen großen Unterschied im Ausdruck. Es erscheinen nur die ersten drei Zahlen, 100 bis 102, danach steht nichts mehr im ganzen Block, bis der Inhalt von 166 die Endzahl 190 erreicht hat. Erst danach, nach dem Loslaufen des Kassettenmotors und dem Einlesen des nächsten Schubes, erscheinen die nächsten drei Zahlen.

Schlußfolgerung:

Durch POKEn der Zahl 191 in die Speicherzelle 166 zu einem beliebigen Zeitpunkt können wir sowohl beim Abspeichern, als auch beim Einlesen einer Datei dem Computer vorgaukeln, der Kassetten-Puffer sei bereits abgearbeitet. Dadurch wird der Kassettenmotor eingeschaltet und der nächste Schub ein- beziehungsweise ausgelesen.

Texteinschub #3 Fehlererkennung mit Parity-Bits

Bei der Datenübertragung zwischen Peripheriegeräten, insbesondere zwischen Datasette und dem Computer kommt es recht häufig vor, daß Fehler auftreten. Diese Fehler haben alle möglichen Ursachen und trotz aller Anstrengungen der Ingenieure lassen sie sich leider nicht völlig vermeiden.

An besonderen Schwachstellen werden daher Maßnahmen getroffen, um Fehler wenigstens zu erkennen und Programme abzubrechen, bevor größerer Schaden entsteht. Die mißlichen »LOAD ERROR«-Meldungen sprechen da eine deutliche Sprache. Die einfachste Art, Fehler zu erkennen — ich sollte genauer sagen: einzelne Bitfehler zu erkennen — geschieht über sogenannte »Parity-Bits«. Die Methode besteht darin, daß zu einem Datenwort, zum Beispiel einem Byte, ein zusätzliches Bit hinzugefügt wird und zwar so, daß die Quersumme immer eine gerade oder auch eine ungerade Zahl ergibt.

Bevor ein Wort übertragen wird, errechnet der Sender das Parity-Bit und fügt es dem Wort als zusätzliches Bit hinzu. Der Empfänger, der diese Prüfmethode natürlich auch kennen muß, rechnet die Quersumme aus. Wenn sie stimmt, nimmt er das Parity-Bit weg und arbeitet mit dem richtigen Wort weiter. Wenn die Quersumme nicht stimmt, schlägt er Alarm.

Sie werden sicher schon bemerkt haben, daß in meinem Beispiel natürlich ein Doppelfehler, nämlich zwei falsche Bits, natürlich nicht erkannt werden. Um das zu erreichen, müßte man zwei Parity-Bits einführen. Sie sehen natürlich auch, wohin das letztlich führt, nämlich zu einer Vergrößerung der Wortlänge. Man nennt das auch »Redundanz«, vielleicht haben Sie dieses Wort schon einmal gehört. Nun, da gibt es für jeden Anwendungsfall ein Optimum, abhängig von der Wahrscheinlichkeit, welche Art von Fehlern in welcher Häufigkeit auftreten. Im Extremfall gibt es Codiersysteme — zu denen die Parity-Bit-Methode auch gehört — welche in der Lage sind, Fehler nicht nur zu erkennen, sondern gleich zu korrigieren.

Logeleien (Teil 3)

Diese letzte Folge der Logeleien wird zum krönenden Abschluß unseren Commodore 64 als »Schlußfolgerungs-Maschine« vorstellen und den Zusammenhang mit der sogenannten Künstlichen Intelligenz beleuchten.

Das Aschenputtel unseres Basic-Sprachvorrates, den WAIT-Befehl, hatten wir in der letzten Folge untersucht. Dabei kam heraus, daß er — mit nur einem Argument versehen — testet, ob in einer spezifizierten Speicherstelle ein bestimmtes Bit gesetzt ist. Im Gegensatz dazu prüft WAIT mit zwei Argumenten, ob ein bestimmtes Bit gelöscht ist.

16. Nochmal WAIT:
Diesmal mit zwei Argumenten.

Wie funktioniert das? Gehen wir aus vom Befehl

WAIT C,A,B

Dabei ist C die abzufragende Speicherstelle, A eine AND-Maske und B eine EOR-Maske. Sogar im Programmers Reference Guide findet man eine falsche Funktionsbeschreibung. Tatsächlich geschieht folgendes:

1) Der Wert der Speicherstelle C wird gelesen.
2) Dieser Wert wird mit B als EOR-Maske exklusiv-oder-verknüpft
3) Das Ergebnis davon wird mit A als AND-Maske verknüpft.

Anhand eines Beispiels wollen wir uns ansehen, was passiert. Der Befehl

WAIT 1,32,32

hält ein Programm an, bis eine Datasettentaste gedrückt wird (H. Hauck, 64'er, Ausgabe 11/1984, Seite 135). Bit 4 der Speicherstelle 1 ist in dem Fall nämlich Null. Der normale Inhalt dieses Registers ist 55 (binär 0011 0111), also Bit 4 = 1. Die Zahl 32 lautet binär

0001 0000

Mit WAIT1,32,32 geschieht folgendes:

keine Taste:

```
 0011 0111    55
 0001 0000    32
__________ EOR
 0010 0111
 0001 0000    32
__________ AND
 0000 0000     0
```

Das Ergebnis ist Null, der Computer wartet weiter.

Taste gedrückt:

```
 0010 0111    39
 0001 0000    32
__________ EOR
 0011 0111
 0001 0000    32
__________ AND
 0001 0000    32
```

Das Ergebnis ist ungleich Null, der Computer geht im Programm weiter.

Im Programmers Reference Guide steht es genau umgekehrt: Danach soll erst die AND und dann die EOR-Operation stattfinden. Lassen Sie uns diese Variante ebenfalls einmal durchspielen:

keine Taste:

```
 0011 0111    55
 0010 0000    32
__________ AND
 0010 0000
 0010 0000    32
__________ EOR
 0000 0000     0
```

Ergebnis also wieder Null. Der Computer wartet weiter.

Taste gedrückt:

```
 0010 0111    39
 0010 0000    32
__________ AND
 0010 0000
 0010 0000    32
__________ EOR
 0000 0000     0!
```

Auch wenn eine Taste gedrückt ist, würde bei dieser Funktionweise der Computer weiter warten bis zum jüngsten Gericht.

Wir sollten uns das merken: Es kommt bei mehreren logischen Verknüpfungen auch auf die Reihenfolge an.

Daß der WAIT-Befehl so selten benutzt wird, hängt zum einen sicherlich mit seiner relativ komplexen Handhabung zusammen, zum anderen aber auch damit, daß meist ein äußeres Ereignis eintreten muß, um den zu überprüfenden Bit-Wert zu verändern. Dafür kommen Kontrollregister in Frage, die für Ein-/Ausgabe-Operationen jeglicher Art eine Rolle spielen, einige VIC-II-Chip-Register und auch Speicherstellen des SID-Chip. Die beiden letzteren sind allerdings von der Firmware her ohnehin auf allerlei Ereignisse hin programmiert, so daß eine Behandlung mittels des WAIT-Befehls lediglich manchmal als Alternative gesehen werden kann. Bleiben also die CIA-Register, de-



ren komplexe Funktion aber meist eher zur Bearbeitung in Assembler zwingt als in Basic.

Ein Anwendungsbeispiel soll aber noch vorgestellt werden. Es stammt von H. Kohlen und wurde im 64'er, Ausgabe 10/1984, Seite 92 abgedruckt. Es handelt sich um eine Joystickabfrage (Port 2), die allerdings immer nur eine Joystickbewegung erfaßt:

WAIT56320,16,16 wartet auf Feuerknopf
WAIT56320,4,4 wartet auf Linksbewegung
WAIT56320,1,1 wartet auf Hochbewegung
WAIT56320,2,2 wartet auf Abwärtsbewegung
WAIT56320,8,8 wartet auf Rechtsbewegung

Das gleiche mit der Speicherstelle 56321 bezieht sich dann auf den Joystickport 1. Wenn also ein Programmabschnitt warten soll, bis der Feuerknopf gedrückt wird, ist WAIT56320,16,16 sicher eine Möglichkeit, so etwas zu programmieren.

17. Logische Sätze

Im Grunde haben wir bisher lediglich das logische Handwerkszeug kennengelernt. Zwar erbrachte das auch einige direkte Anwendungen, aber richtig logische Probleme zu lösen, dazu bedarf es noch einiger Überlegungen. Genau das wollen wir versuchen: Kann ein Computer verzwickte logische Aufgabenstellungen befriedigend lösen?

Die meisten logischen Fragestellungen bestehen nicht nur aus der Anwendung einer einzigen Verknüpfungsart, sondern — wie schon der WAIT-Befehl mit zwei Argumenten — aus mehreren verkoppelten Aussagen. Wie stellt man die Wahrheitswerte solcher Kombinationen her?

Zunächst ein Beispiel für solch einen logischen SATZ (so nennt man diese verkoppelten Aussagen auch häufig):
NOT(A AND (NOTB))

Zum Ergebnis gelangt man wieder über eine Wahrheitstabelle. Abhängig von der Anzahl der Variablen (hier also zwei: A,B) muß eine Anzahl von Zeilen vorgesehen werden, die alle möglichen W/F-Kombinationen enthält. Bei zwei Variablen sind das dann vier Zeilen, allgemein aber bei N Variablen 2^N Zeilen. Die Anzahl der Spalten ergibt sich aus der Menge der Variablen und der vorzunehmenden Verknüpfungen. In Bild 1 sehen Sie die Wahrheitstabelle für unser Beispiel.

Links finden Sie die Kombinationen der Wahrheitswerte der Variablen A und B. Daneben wurde NOTB entwickelt, dann (wie bei algebraischen Rechnungen Schritt für Schritt) die AND-Verknüpfung und schließlich die Verneinung der Spalte davor. Diese letzte Spalte ist auch eine Zusammenstellung der Ergebnisse.

A	B	NOT B	A AND (NOT B)	NOT (A AND (NOT B))
W	W	F	F	W
W	F	W	W	F
F	W	F	F	W
F	F	W	F	W

Bild 1. So sieht die Wahrheitstabelle des logischen Satzes NOT(A AND (NOTB)) aus

A	B	A IMP B
W	W	W
W	F	F
F	W	W
F	F	W

Bild 2. Das ist die Wahrheitstabelle der IMP-Verknüpfung

Das kann auch unser C 64 für uns rechnen. Das Programm »Logelei-1« erledigt das in der Weise, daß es erst nach der logischen Funktion fragt, deren Wahrheitstabelle berechnet werden soll, sich dann selbst verändert (Zeilen 100-130) und schließlich die Berechnungen durchführt.

Zu beachten ist, daß »Logelei-1« nur die normalerweise in unserem Basic-Sprachvorrat enthaltenen logischen Operatoren kennt.

Zwei Sonderfälle von logischen Sätzen sollen Sie kennenlernen. Zunächst die sogenannte »Tautologie« (hochdeutsch: Gleichbedeutung). Davon spricht man, wenn sämtliche Wahrheitswerte des Ergebnisses W (wahr) sind (beziehungsweise —1). Das ist der Fall in Hamlets berühmten Satz (wenn er damit ein inklusiv-Oder gemeint hat) »sein oder nicht sein«. Probieren Sie mal aus:
A OR (NOT A)

(Das Programm Logelei-1 ist übrigens darauf nicht eingerichtet). Das Gegenteil der Tautologie ist die »Kontradiktion« (zu hochdeutsch: Widerspruch). Sämtliche Wahrheitswerte des Ergebnisses sind F (falsch oder 0). Das findet man zum Beispiel bei:
A AND (NOT A)

Als Beispiel sei A die Aussage »es regnet«. Dann ist es eine Kontradiktion zu behaupten: »Es regnet und es regnet nicht«.

Bisher fehlen uns aber noch einige entscheidende Werkzeuge: Wir können noch keine Schlußfolgerungen ziehen. Das steuern wir nun an.

Es gibt in einigen Basic-Dialekten den Operator EQV. Das kommt von »äquivalent«, also »gleich«. Wir dürfen in unserem C 64 getrost dafür das Gleichheitszeichen (=) verwenden. Äquivalenz zweier logischer Sätze liegt vor, wenn die Wahrheitswerte der Ergebnisse gleich sind. Probieren Sie mal mittels unseres Programmes:
a) NOT (A AND B)
b) (NOT A) OR (NOT B)

Beide Sätze sind äquivalent, weil die Ergebnisse gleiche Wahrheitswerte besitzen. Nebenbei: Das ist eines der sogenannten Gesetze von DeMorgan. Das Programm Logelei-1 zeigt sich — wie Sie nun sehen werden — auch schon als »Beweismaschine«. Bei der Frage nach der Funktion geben Sie doch mal dieses DeMorgan-Gesetz ein:
(NOT(AANDB))=((NOTA)OR(NOTB))

Das Ergebnis zeigt lauter -1, also W. Für alle Variablenkombinationen ist dieser Satz also richtig. Vielleicht ahnen Sie jetzt schon, was ein Computer in Fragen Logik zu leisten imstande ist. Probieren Sie doch mal selbst einfach ein paar Äquivalenzen aus. So können Sie selbst noch zum Entdecker werden.

18. Bedingungen

Auf dem Weg zum Computer als »Schlußfolgerungs-Maschine« fehlt uns noch etwas Wichtiges: Bedingungen. Das sind Verknüpfungen der Form »Wenn... dann...«. Man nennt diese Verknüpfungen »Implikationen« und findet in manchen Basic-Dialekten dafür den Operator IMP. Wir dürfen statt dessen >= verwenden.

(Der Vollständigkeit halber sei noch angemerkt, daß wir im C 64 anstelle von EOR »<>« verwenden können. Der C 128 kennt diesen Befehl als XOR.)

Im weiteren soll für diese Wenn-/Dann-Verknüpfung immer IMP verwendet werden. Sie können dann in einem Programm leicht >= dafür einsetzen. Für eine Aussagen-Operation der Form A IMP B gibt es wieder genaue Vorschriften, wie die einzelnen Wahrheitswerte miteinander verrechnet werden. Bild 2 zeigt Ihnen die Wahrheitstabelle der IMP-Operation:

Nur dann also, wenn A richtig und B falsch ist, ist auch A IMP B falsch. Diese Wahrheitstabelle erstellt Ihnen auch ohne weiteres Logelei-1. Wenn Sie schon am Probieren sind, dann geben Sie doch noch mal diese Äquivalenz ein:
(A>=B)=((NOTA)ORB)

Das Ergebnis zeigt: Anstelle von A IMP B kann auch (NOTA) ORB verwendet werden. Solche Äquivalenzen sind sehr brauchbar. Deshalb werden wir uns noch eine weitere IMP-Operation ansehen. Überprüfen Sie doch mal mittels Logelei-1 die Wahrheitswerte dieses Satzes:
(NOTB) IMP (NOTA)

Sie sehen, daß genau dieselben Wahrheitswerte herauskommen wie für A IMP B. Auch diese Sätze sind demnach äquivalent. Man nennt den letzteren »Kontraposition« von A IMP B. Als Beispiel für eine Anwendung dieser Erkenntnis wollen wir einen Beweis führen:

Der Satz »Wenn x^2 ungerade ist, dann ist x ungerade« soll bewiesen werden. Wenn wir für A einsetzen »x^2 ungerade« und für B »x ungerade«, dann ist also zu beweisen A IMP B. Die Kontraposition davon wäre dann der Satz: »Wenn x gerade ist, dann ist auch x^2 gerade«. Das zu zeigen, ist relativ einfach. Eine gerade Zahl x ist auszudrücken durch 2n: x = 2n. Dann ist für x^2 zu schreiben:
$x^2 = (2n)(2n) = 2(2n^2)$,
was wieder eine gerade Zahl ist. Damit haben wir die Kontraposition bewiesen. Weil aber die Kontraposition äquivalent ist zu unserem Ausgangssatz, ist so auch dieser bewiesen.

Was damit zu zeigen war, ist die Tatsache, daß man zu Beweisen häufig nur einen geeigneten äquivalenten Satz beweisen muß, was unter Umständen sehr viel leichter sein kann.

19. Argumente, Schlußfolgerungen

Wenn jemand in einer Diskussionsrunde einen anderen Teilnehmer abbügelt, indem er behauptet, dieser habe keine gültigen Argumente, dann kommt er dem, was ein Logiker unter diesem Begriff versteht, schon ziemlich nahe. Für diesen ist ein Argument eine Beziehung zwischen logischen Sätzen, die er »Prämissen« (Voraussetzungen) nennt und einem Satz, der »logischer Schluß« heißt. Solche Argumente können dann »gültig« sein oder »irrig« (also ungültig).

Gültigkeit liegt nur dann vor, wenn alle Prämissen den Wahrheitswert W aufweisen und außerdem gleichzeitig auch der logische Schluß den Wahrheitswert W hat. Das ist vielleicht so

trocken etwas unverständlich. Wir sehen uns daher ein Beispiel an.

Unser Beispiel beinhaltet ein grundlegendes Gesetz, das Gesetz der logischen Schlußfolgerung (Gesetz der Syllogistik, aufgestellt von Aristoteles). Man kann es so ausdrücken: »Aus A folgt B. Aus B folgt C. Dann folgt auch aus A die Aussage C.« Die beiden ersten Sätze sind die Prämissen. Der letzte Satz ist der logische Schluß. In Formeln:
Prämissen: A IMP B
B IMP C
Schluß: A IMP C

Das Ganze ist ein Argument, und wir müssen nun nachprüfen, ob es gültig ist. Dazu bauen wir uns eine Wahrheitstabelle, die nun (weil drei Variable vorhanden sind) $2^3 = 8$ Zeilen enthalten muß. Außerdem soll jedem Satz und jeder Variablen eine Spalte zugeordnet sein. In Bild 3 ist diese Wahrheitstabelle zu sehen.

A	B	C	A IMP B	B IMP C	A IMP C
W	W	W	W	W	W
W	W	F	W	F	F
W	F	W	F	W	W
W	F	F	F	W	F
F	W	W	W	W	W
F	W	F	W	F	W
F	F	W	W	W	W
F	F	F	W	W	W

Bild 3. Wahrheitstabelle eines gültigen Argumentes. Hier das syllogistische Gesetz des Aristoteles.

Links finden Sie alle möglichen Kombinationen der Variablen-Wahrheitswerte, rechts daneben dann die Werte, die bei den drei Sätzen herauskommen. In den markierten Zeilen sind die Werte beider Prämissen W und — wie Sie unschwer feststellen werden — auch der Wahrheitswert des Schlusses ist dort W. Damit ist dieses Argument gültig.

Eine andere, etwas computergerechtere Form, lernen wir nun noch kennen. Dabei verknüpft man alle Prämissen durch die AND-Operation und dies so entstandene Gebilde wird durch die IMP-Operation mit dem Schluß verkoppelt. Das Ergebnis ist eine Tautologie, wenn das Argument gültig ist. Unser Beispiel lautet in dieser Form dann:
((A IMP B)AND(B IMP C))
IMP (A IMP C)

Unser Computer wird also eine »Schlußfolgerungs-Maschine«, wenn wir das Programm Logelei-1 etwas erweitern und umbauen. Als »Logelei-2« sehen Sie diese Schlußfolgerungsmaschine hier abgedruckt.

Im Grunde genommen tut dieses Programm nichts anderes als Logelei-1, nur sind jetzt unterschiedliche Variablen-Anzahlen zugelassen (von einer bis zu vier Variablen). Ebenso wie vorhin wird hier auch nach Anzeigen der vorhandenen Funktion gefragt, ob eine andere gewünscht sei. Danach muß die Anzahl der Variablen eingegeben werden und unter Umständen die neue Funktion. Wenn eine Schlußfolgerung zu untersuchen ist, wird sie in der eben erklärten Form (Prämissen AND-verknüpft etc.) eingegeben. Ansonsten verfahren Sie wie bei Logelei-1. Der Computer erstellt nun eine Wahrheitstabelle. Liegt eine Tautologie vor, dann sind alle Ergebnis-Wahrheitswerte gleich -1 und damit das Argument gültig, die Schlußfolgerung also richtig. Um das Programm überschaubar zu halten, sind einige Eleganzen nicht verwendet worden. Aber vielleicht reizt Sie das ja zum Erweitern, Verfeinern, etc.

Versuchen Sie es mal mit dem obigen Beispiel. Sie sehen, wir können mit Logelei-2 ein logisches Gesetz beweisen. Natürlich kann man diese Schlußfolgerungsmaschine auch im alltäglichen Bereich einsetzen. Das soll das folgende einfache Beispiel zeigen:

Ein Student erzählt seine Situation: »Wenn ich jobbe, dann kann ich nicht gleichzeitig studieren. Wenn ich studiere, dann schaffe ich das Examen. Wenn ich nicht jobbe, dann muß ich am Hungertuch nagen. Ich habe mich entschlossen, trotzdem zu studieren. Dann habe ich mein Examen und nage am Hungertuch.« Das sind 4 Sätze mit 4 Variablen:
A = ich jobbe
B = ich studiere
C = ich schaffe das Examen
D = ich nage am Hungertuch

In der verkürzten Form lautet die Äußerung des Studenten dann:
Prämissen: A IMP NOTB
B IMP C
NOTA IMP D
B
Schluß: C AND D

Nach dem Starten der Schlußfolgerungs-Maschine geben Sie also ein:
((A IMP (NOTB)) AND (B IMP C) AND ((NOTA) IMP C) AND B) IMP (C AND D)

Anstelle von IMP verwenden Sie die oben genannten Zeichen. Die Zwischenräume können wegbleiben, denn sie sind nur der Deutlichkeit halber eingefügt. Außerdem sagen Sie bitte dem Computer, daß er es mit vier Variablen zu tun hat. Das Ergebnis zeigt, daß die Schlußfolgerung richtig ist: Lauter -1 treten auf.

So, nun können Sie dieses Programm auf eigene Probleme anwenden. Ein paar nette Beispiele finden Sie in einem Artikel von D. Herrmann in der Zeitschrift computer colleg, Ausgabe 2/1985, Seite 26. Wenn Sie öfters versuchen, Schlüsse auf diese Weise per Computer zu überprüfen, werden Sie feststellen, daß der schwierigste Schritt die Umwandlung von Sätzen in die mathematische Kurzform darstellt. Das kennen Sie vielleicht schon von Textaufgaben in der Mathematik her.

20. Künstliche Intelligenz?

Wenn Sie diese Folge bis hierher gelesen und vielleicht einige Schlüsse selbst ausprobiert haben, ist bei Ihnen möglicherweise der Eindruck entstanden, daß Sie hier so etwas wie den Keim der Künstlichen Intelligenz (auch mit dem Kürzel KI bedacht) vorliegen haben. Ein Programm, das in der Lage ist, mehr Variable — vielleicht sogar Hunderte — zu verarbeiten, welches die Schlüsse aus allen möglichen Variablenkombinationen selbst erstellt und mittels der Tautologien überprüft auf Richtigkeit, solch ein Programm wäre im Prinzip auf der gleichen Basis wie »Logelei-2« zu erstellen, also relativ einfach. Zwar hätte es ziemlich große Ausmaße und es bedürfte sicherlich eines wesentlich schnelleren Prozessors, um in absehbaren Zeiten zu Resultaten zu kommen, aber es wäre technisch realisierbar. Ist das Künstliche Intelligenz, und — diese etwas utopisch anmutende Zusatzfrage stellt sich zwangsläufig — wäre das Programm nicht in der Lage, Firmen, Gesellschaften, ja auch dem Staat eine wichtige Hilfe bei Entscheidungen zu sein?

Versuchen wir zunächst, die Frage nach KI zu beantworten, dann erkennen wir sehr schnell, daß eine Antwort nicht möglich ist. Noch gibt es keine allgemein anerkannte Definition, was Intelligenz eigentlich sei. Dann kann auch kein Begriff einen sinnvollen Inhalt haben, der daraus abgeleitet ist.

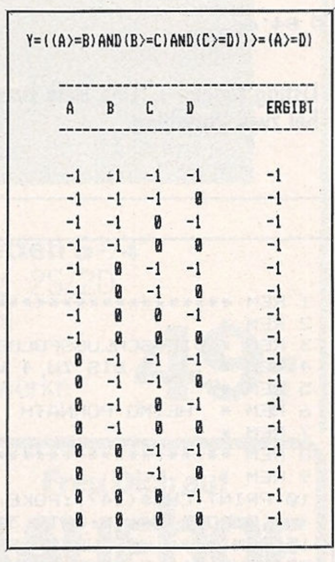
Die zweite Frage ist nicht ganz so leicht zu klären. Solch eine »Super-Schlußfolgerungs-Maschine« wäre sicherlich in einigen Anwendungsbereichen sehr nützlich. Aber vielleicht ist Ihnen schon bei der Formulierung von Problemstellungen für unser Programm »Logelei-2« aufgefallen, daß es da manchmal Schwierigkeiten gibt. Irgendwie fällt es ab und zu schwer, einen Satz, in dem Adjektive oder Adverbien eine große Rolle spielen, in die Kurzform umzusetzen. Nun haben Sie recht, wenn Sie sagen, daß das sicher eine Frage der Feinheit der Methode sei. Wie haben hier ja auch nur einen kleinen Zipfel der Logik zu fassen bekommen. Aber trotzdem bleiben Probleme. Im täglichen Leben werden Sie höchst selten Situationen vor sich haben, die mit JA oder NEIN (Wahr oder Falsch) zu bewältigen sind. Da gibt es zum Beispiel noch ein JA, ABER oder ein VIELLEICHT oder ein JEIN etc. Die zweiwertige Logik, auf der all das fußt, was wir in diesen drei Folgen behandelt haben, spielt in der Realität nur eine untergeordnete Rolle! Sie ist deshalb auch nur höchst selten und jedesmal mit einer gehörigen Portion Skepsis auf reale Problemstellungen anwendbar.

Allen, die neugierig auf dieses Thema geworden sind, möchte ich zwei Bücher ans Herz legen, die ebenso unterschiedlich wie nützlich sind:

1) Seymour Lipschutz: »Essential Computer Mathematics«. McGraw-Hill 1982. ISBN 0-07-037990-4. Das ist einer der Bände der Serie »Schaum's Outline Series«, der auch noch andere mathematische Themen anpackt und das alles mit vielen Beispielen würzt.

2) Douglas R. Hofstadter: »Gödel, Escher, Bach ein Endloses Geflochtenes Band«. Klett-Cotta 1985. ISBN 3-608-93037-X. Das ist für jeden, der sich für Computer, Logik, KI und das Denken an sich interessiert, eine Art Bibel.

(Heimo Ponnath/gk)



Y=((A>=B)AND(B>=C)AND(C>=D))>=(A>=D)

A	B	C	D	ERGIBT
-1	-1	-1	-1	-1
-1	-1	-1	0	-1
-1	-1	0	-1	-1
-1	-1	0	0	-1
-1	0	-1	-1	-1
-1	0	-1	0	-1
-1	0	0	-1	-1
-1	0	0	0	-1
0	-1	-1	-1	-1
0	-1	-1	0	-1
0	-1	0	-1	-1
0	-1	0	0	-1
0	0	-1	-1	-1
0	0	-1	0	-1
0	0	0	-1	-1
0	0	0	0	-1

Bild 4. Das Programm »Logelei-2« überprüft Aussagen. Wenn alle Ergebnisse —1 lauten, ist die Aussage richtig (so wie hier).

```
 1 REM ******************************    <132>
 2 REM *                            *    <051>
 3 REM *    LOGELEIEN FUER ZWEI     *    <224>
 4 REM *         VARIABLE           *    <026>
 5 REM *                            *    <054>
 6 REM *  HEIMO PONNATH  HAMBURG 1985  * <131>
 7 REM *                            *    <056>
 8 REM ******************************    <139>
 9 REM                                   <071>
 10 PRINT CHR$(147):POKE 53280,0:POKE 53281
 ,0:POKE 646,5:GOTO 30                   <086>
 15 REM **** UP CURSOR SETZEN ****       <067>
 20 POKE 211,S:POKE 214,Z:SYS 58640:RETURN <220>
 25 REM **** HP-1: FUNKTIONENEINGABE *** <148>
 30 S=11:Z=5:GOSUB 20:POKE 646,7:PRINT"LOGE
 LEIEN FUER ZWEI"                        <076>
 35 S=16:Z=6:GOSUB 20:POKE 646,2:PRINT"VARI
 ABLE"                                   <227>
 40 S=6:Z=10:GOSUB 20:POKE 646,5:PRINT"IM P
 ROGRAMM IST DIE FUNKTION:"              <136>
 45 K=1:GOSUB 100:K=0:S=0:Z=15:GOSUB 20:PRI
 NT"C="L$                                <028>
 50 S=6:Z=20:GOSUB 20                    <188>
 55 PRINT CHR$(18)"A"CHR$(146)"LTE ODER "CH
 R$(18)"N"CHR$(146)"EUE FUNKTION ?"      <227>
 60 GET A$:IF A$<>"A"AND A$<>"N"THEN 60  <218>
 65 IF A$="A"THEN 95                     <055>
 70 S=0:Z=22:GOSUB 20:PRINT"C(A,B)=";:INPUT
   L$:POKE 781,22:SYS 59903              <176>
 75 GOSUB 20:PRINT"C="L$:FOR I=0 TO 500:NEX
 T:POKE 646,0                            <002>
 80 PRINT CHR$(147)CHR$(17):PRINT"100L$="CH
 R$(34)L$CHR$(34)                        <182>
 85 PRINT"110DEFFNL(A)="L$:PRINT"120DEFFNM(
 B)="L$:PRINT"RUN95"CHR$(19);            <023>
 90 POKE 631,13:POKE 632,13:POKE 633,13:POK
 E 634,13:POKE 198,4:END                 <195>
 94 REM **** HP-2: BERECHNUNGEN ****     <056>
 95 POKE 646,5:PRINT CHR$(147)           <227>
 100 L$="NOT(A AND(NOT B))"              <206>
 110 DEF FN L(A)=NOT(A AND(NOT B))       <020>
 120 DEF FN M(B)=NOT(A AND(NOT B))       <110>
 130 IF K=1 THEN RETURN                  <139>
 140 S=5:Z=5:GOSUB 20:POKE 646,2:PRINT"HIER
   IST DIE WAHRHEITSTABELLE:"            <230>
 145 PRINT:PRINT:PRINT:POKE 646,1:PRINT TAB
  (5)"A"TAB(10)"B"TAB(25)"ERGIBT C"      <085>
 150 PRINT TAB(4)"-----------------------
 -------":POKE 646,5:PRINT               <215>
 155 FOR A=-1 TO 0:FOR B=-1 TO 0:PRINT TAB(
 5)A TAB(10)B TAB(27)FN M(B)             <078>
 160 NEXT B:NEXT A                       <255>
 165 PRINT:POKE 646,1:PRINT TAB(4)"--------
 -------------------------":POKE 646,5   <227>
 170 S=0:Z=20:GOSUB 20:PRINT"C="L$       <035>
```

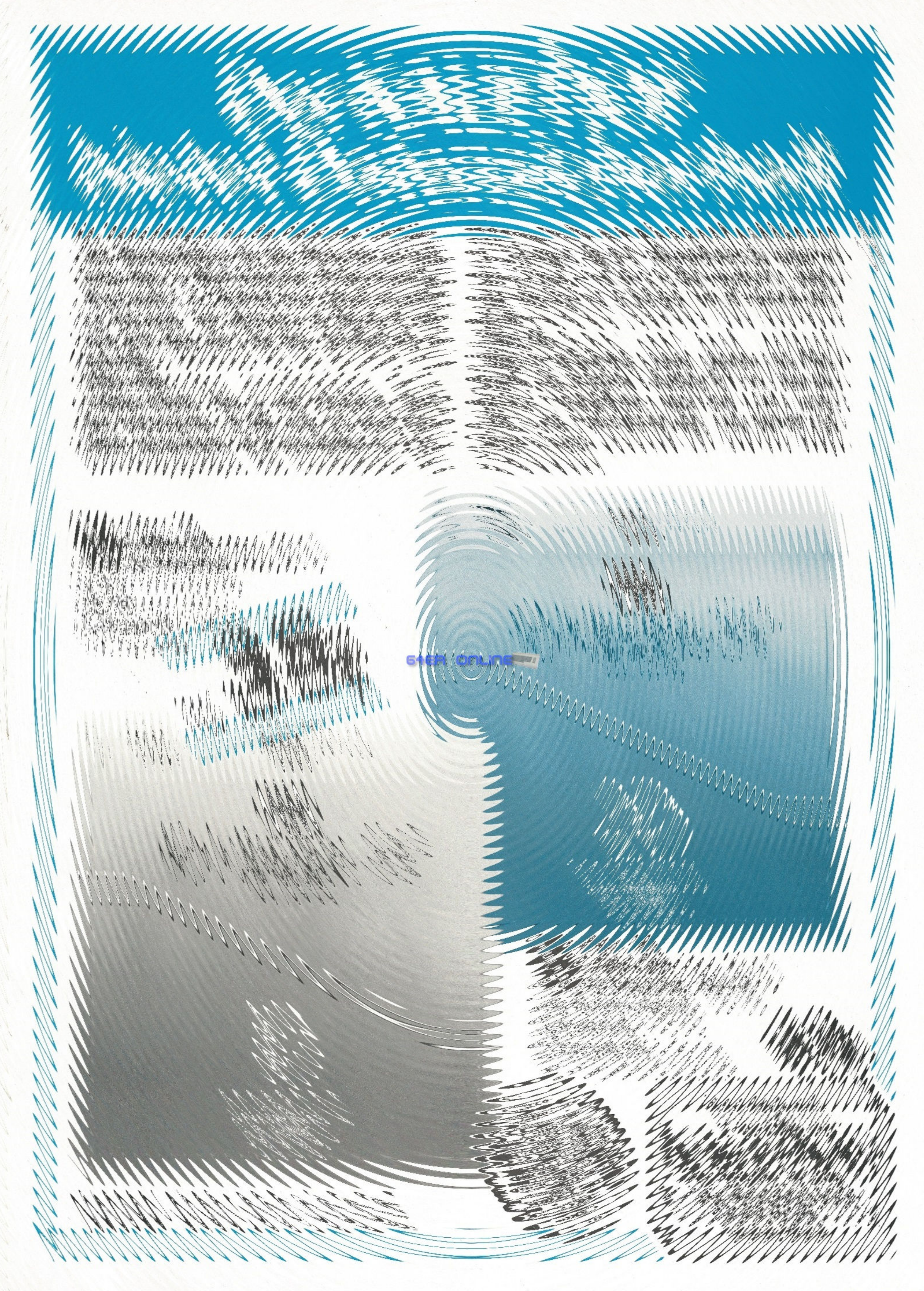
© 64'er

Listing Logelei-1. Eine Hilfe zum Erstellen von Wahrheitstabellen bei zwei Variablen

```
 1 REM ******************************    <132>
 2 REM *                            *    <051>
 3 REM * DIE SCHLUSSFOLGERUNGSMASCHINE * <217>
 4 REM *       BIS ZU 4 VARIABLE    *    <062>
 5 REM *                            *    <054>
 6 REM *  HEIMO PONNATH  HAMBURG 1985  * <131>
 7 REM *                            *    <056>
 8 REM ******************************    <139>
 9 REM                                   <071>
 10 PRINT CHR$(147):POKE 53280,0:POKE 53281
 ,0:POKE 646,5:GOTO 30                   <086>
 15 REM **** UP CURSOR SETZEN ****       <067>
 20 POKE 211,S:POKE 214,Z:SYS 58640:RETURN <220>
 25 REM **** HP-1: FUNKTIONENEINGABE *** <148>
 30 S=7:Z=5:GOSUB 20:POKE 646,7:PRINT"SCHLU
 SFOLGERUNGSMASCHINE                     <005>
 35 S=10:Z=6:GOSUB 20:POKE 646,2:PRINT"BIS
 ZU 4 VARIABLE"                          <119>
 40 S=6:Z=9:GOSUB 20:POKE 646,5:PRINT"IM PR
 OGRAMM IST DIE FUNKTION:"               <126>
 45 K=1:GOSUB 100:K=0:S=0:Z=12:GOSUB 20:PRI
 NT"Y="L$                                <225>
 46 Z=14:GOSUB 20:PRINT"MIT "N" VARIABLEN" <113>
 50 S=6:Z=16:GOSUB 20                    <199>
 55 PRINT CHR$(18)"A"CHR$(146)"LTE ODER "CH
 R$(18)"N"CHR$(146)"EUE FUNKTION ?"      <227>
 60 GET A$:IF A$<>"A"AND A$<>"N"THEN 60  <218>
 62 Z=17:GOSUB 20:PRINT"WIEVIELE VARIABLE (
 MAX 4)";:INPUT B$                       <092>
 63 N=VAL(B$):IF N<1 OR N>4 THEN 62      <140>
 65 IF A$="A"THEN 95                     <055>
 70 S=0:Z=19:GOSUB 20:PRINT"Y(A,B,..)=";:IN
 PUT L$:POKE 781,19:SYS 59903            <181>
 75 GOSUB 20:PRINT"Y="L$:FOR I=0 TO 500:NEX
 T:POKE 646,0                            <024>
 80 PRINT CHR$(147)CHR$(17):PRINT"100L$="CH
 R$(34)L$CHR$(34)                        <182>
 85 PRINT"110DEFFNL("CHR$(64+N)")="L$:PRINT
 "120N="N:PRINT"RUN95"CHR$(19);          <254>
 90 POKE 631,13:POKE 632,13:POKE 633,13:POK
 E 634,13:POKE 198,4:END                 <195>
 94 REM **** HP-2: BERECHNUNGEN ****     <056>
 95 POKE 646,5:PRINT CHR$(147)           <227>
 100 L$="((A>=B)AND(B>=C)AND(C>=D))>=(A>=D)
 "                                       <080>
 110 DEF FN L(D)=((A>=B)AND(B>=C)AND(C>=D))
 >=(A>=D)                                <018>
 120 N= 4                                <137>
 125 C$="{4SPACE}---------------------------
 -------"                                <146>
 130 IF K=1 THEN RETURN                  <139>
 140 S=2:Z=0:GOSUB 20:POKE 646,2:PRINT"HIER
   IST DIE WAHRHEITSTABELLE VON:"        <247>
 145 S=0:Z=2:GOSUB 20:PRINT"Y="L$:POKE 646,
 1:PRINT:PRINT C$:POKE 646,5             <025>
 150 ON N GOSUB 200,300,400,500:PRINT:POKE
 646,1:PRINT C$:POKE 646,5:PRINT         <207>
 155 PRINT"WENN ALLE ERGEBNISSE -1 LAUTEN,
 DANN"                                   <193>
 160 PRINT"IST DER SCHLUSS RICHTIG ! (TASTE
 )":POKE 198,0:WAIT 198,1                <202>
 190 END                                 <192>
 195 REM ***** 1 VARIABLE *****          <055>
 200 PRINT TAB(15)"A"TAB(30)"ERGIBT":POKE 6
 46,1:PRINT C$:PRINT:POKE 646,5          <090>
 205 FOR A=-1 TO 0:PRINT TAB(15)A TAB(30)FN
  L(A)                                   <173>
 210 NEXT A:RETURN                       <090>
 295 REM ***** 2 VARIABLE *****          <158>
 300 PRINT TAB(5)"A"TAB(10)"B"TAB(30)"ERGIB
 T":POKE 646,1:PRINT C$:PRINT:POKE 646,
 5                                       <008>
 305 FOR A=-1 TO 0:FOR B=-1 TO 0         <171>
 310 PRINT TAB(5)A TAB(10)B TAB(30)FN L(B) <092>
 315 NEXT B:NEXT A:RETURN                <071>
 395 REM ***** 3 VARIABLE *****          <003>
 400 PRINT TAB(5)"A"TAB(10)"B"TAB(15)"C"TAB
 (30)"ERGIBT"                            <172>
 405 POKE 646,1:PRINT C$:PRINT:POKE 646,5 <070>
 410 FOR A=-1 TO 0:FOR B=-1 TO 0:FOR C=-1 T
 O 0                                     <022>
 415 PRINT TAB(5)A TAB(10)B TAB(15)C TAB(30
 )FN L(C)                                <030>
 420 NEXT C:NEXT B:NEXT A:RETURN         <202>
 495 REM ***** 4 VARIABLE *****          <104>
 500 PRINT TAB(5)"A"TAB(10)"B"TAB(15)"C"TAB
 (20)"D"TAB(30)"ERGIBT"                  <223>
 505 POKE 646,1:PRINT C$:PRINT:POKE 646,5 <170>
 510 FOR A=-1 TO 0:FOR B=-1 TO 0:FOR C=-1 T
 O 0:FOR D=-1 TO 0                       <132>
 515 PRINT TAB(5)A TAB(10)B TAB(15)C TAB(20
 )D TAB(30)FN L(D)                       <231>
 520 NEXT D:NEXT C:NEXT B:NEXT A:RETURN  <024>
```

© 64'er

Listing Logelei-2. Die Schlußfolgerungs-Maschine. Ein Programm zum Überprüfen der Stimmigkeit logischer Schlüsse.




64er online



C 64 extern — Der Weg nach draußen

In diesem Teil des Kurses werden wir uns mit dem Control-Port 1 beschäftigen. Er eignet sich nicht nur zum Anschluß eines Joysticks. Was man noch alles mit ihm machen kann, wollen wir am Beispiel eines Lichtgriffels im Selbstbau zeigen.

Am Control-Port 1 des C 64 gibt es nur noch einen Anschluß, dessen Bedeutung wir noch nicht genau kennen. Gemeint ist Pin 6, der Light Pen-Anschluß, von dem wir im letzten Teil schon etwas gehört haben. Er hat zwei verschiedene Funktionen. Zum einen wird hier die Information über den Zustand des Feuerknopfes eingespeist. Zur anderen interessanten Anwendung kommen wir jetzt.

Vielleicht haben Sie schon einmal von einer Computeranlage gehört (oder eine gesehen), bei der man bestimmte Funktionen mit einem speziellen Stift, der mit dem Computer verbunden ist, direkt am Bildschirm anwählen kann. Solch einen Stift nennt man Lichtgriffel. Ein Lichtgriffel ist sehr einfach an den C 64 oder den VC 20 anzuschließen. Man kann sich den »Lightpen« sogar selbst bauen.

Im Innern eines Lichtgriffels steckt als wichtigstes Element ein Fototransistor, eine Fotodiode oder ähnliches. Dieser Fototransistor reagiert wie unser LDR aus der letzten Folge auf Licht. Das Licht wird vom Elektronenstrahl der Bildröhre erzeugt, wenn dieser auf die Mattscheibe trifft. Der Fototransistor an den Bildschirm gehalten, registriert den Elektronenstrahl als kurzen Lichtimpuls. Das ist auch der Grund, weshalb der Lichtgriffel nur an relativ hellen Bildschirmstellen funktioniert, denn nur hier entsteht ein ausreichender Lichtimpuls. Wenn vom Lichtgriffel ein Impuls an den Computer gegeben wird, prüft dieser, für welchen Punkt des Schirms er gerade das Signal an den Bildschirm geschickt hat. Mit Hilfe dieser Information kann er dann die X- und Y-Koordinate der derzeitigen Position des Lichtgriffels auf dem Bildschirm zur Verfügung stellen. Diese beiden Werte können wir abfragen und für unsere Programme verwenden.

Lichtgriffel selbstgebaut

Bevor wir wieder etwas programmieren, wollen wir uns einen Lichtgriffel beschaffen und an unseren Computer anschließen. Für den Lichtgriffel benötigen wir, wie oben schon gesagt, einen Fototransistor. Es wurde der Typ TIL 78 verwendet. Alle Bauteile, die für diesen Kursteil benötigt werden, sind im Elektronikfachhandel erhältlich. Der LDR aus der letzten Folge reicht für unseren Lichtgriffel nicht aus, weil er zu langsam auf die Lichtschwankungen reagiert und deshalb keine brauchbaren Impulse liefert.

Wenn Sie schon in der glücklichen Lage sind, einen echten Lichtgriffel zu besitzen, können Sie die Versuche selbstverständlich auch mit diesem ausführen. Im folgenden soll jedoch nur auf den Fototransistor eingegangen werden.

Der TIL 78 hat zwei Beine. Sie heißen »Kollektor« und »Emitter«. Nun stellt sich die Frage, welches Bein welche Funktion hat? Es gibt drei Möglichkeiten, dies herauszufinden. Die erste ist für den Elektronikneuling wohl die beste: Sie fragen in dem Fachgeschäft, in dem Sie den Fototransistor kaufen. Zu Hause haben Sie dann zwei weitere Möglichkeiten: Die Seite des Fototransistorgehäuses, auf der sich der Kollektor befindet, ist etwas abgeflacht. Außerdem hat der Kollektor das kürzere Anschlußbein. Die zweite Erkennungsmethode ist allerdings sicherer als die dritte, weil irgend jemand das längere Bein zum kürzeren gemacht haben könnte!

```
100 REM ******************************        <238>
110 REM *                            *        <159>
120 REM *        D E M O   Z U M     *        <119>
130 REM *        ---------------     *        <241>
140 REM *                            *        <189>
150 REM *   L I C H T G R I F F E L  *        <072>
160 REM *   -----------------------  *        <011>
170 REM *                            *        <219>
180 REM *  BY    TOBIAS       NICOL  *        <229>
190 REM *                            *        <239>
200 REM *      NEUWIESENSTRASSE 20   *        <202>
210 REM *                            *        <003>
220 REM *      6000  FRANKFURT  71   *        <042>
230 REM *                            *        <023>
240 REM ******************************        <122>
250 :                                         <226>
260 REM ********* INFORMATION *********       <220>
270 PRINT "{CLR,4DOWN}SOLL DER PUNKT JEWEI
    LS                                        <131>
280 PRINT "{DOWN}WIEDER GELOESCHT WERDEN?{
    DOWN}                                     <247>
290 PRINT "BITTE 'J' ODER 'N' EINGEBEN!       <121>
300 GET A$ : IF A$ = "" THEN 300              <239>
310 IF A$ = "N" THEN L = 1                    <001>
320 :                                         <042>
330 REM ******* INITIALISIERUNG *******       <047>
340 POKE 53280,0 : POKE 53281,1               <218>
350 PRINT "{CLR}" : A = 1024 : B = 40         <238>
360 C = 24 : D = 53267                        <039>
370 :                                         <092>
380 REM ***** FARBSPEICHER SETZEN *****       <206>
390 FOR I = 55296 TO 56295                    <128>
400 POKE I,0 : NEXT I                         <026>
410 :                                         <132>
420 REM ******  WERTE  EINLESEN  ******       <228>
430 REM ****** UND STABILISIEREN ******       <066>
440 FOR I = 1 TO 5                            <030>
450 X = X + PEEK (D)                          <238>
460 Y = Y + PEEK (D+1)                        <028>
470 NEXT I                                    <044>
480 X = INT (X/5) : Y = INT (Y/5)             <175>
490 :                                         <212>
500 REM ** IN KOORDINATEN UMWANDELN  **       <178>
510 KX = INT ((X-50)/180*(B-1))               <154>
520 KY = INT ((Y-45)/190*C)                   <057>
530 :                                         <254>
540 REM ** KOORDINATEN KONTROLLIEREN **       <191>
550 IF KX < 0 THEN KX = 0                     <210>
560 IF KY < 0 THEN KY = 0                     <238>
570 IF KX > B-1 THEN KX = B-1                 <074>
580 IF KY > C THEN KY = C                     <031>
590 :                                         <058>
600 REM **** PUNKT SETZEN/LOESCHEN ****       <064>
610 IF L=0 THEN POKE A+XA+YA*B,32             <078>
620 POKE A+KX+KY*B,160                        <164>
630 :                                         <098>
640 REM ** NEUE WERTE & RUECKSPRUNG  **       <074>
650 XA = KX : YA = KY : X = 0 : Y = 0         <171>
660 GOTO 420                                  <136>
670 :                                         <138>
680 :                                         <148>
690 :                                         <158>
700 REM ******************************        <074>
710 REM *** AENDERUNGEN FUER VC-20  ***       <027>
720 REM ******************************        <094>
730 REM                                       <028>
740 REM SETZEN SIE DIESE ZEILEN EIN:          <070>
750 REM                                       <048>
760 REM 340 POKE 36879,24                     <144>
770 REM 350 PRINT "{CLR}" : A=7680 : B=22     <185>
780 REM 360 C=22 : D=36870                    <217>
790 REM 390 FOR I = 38400 TO 38905            <199>
800 REM 510 KX = INT ((X-55)/95*(B-1))        <211>
810 REM 520 KY = INT ((Y-38)/89*(C))          <028>
820 REM                                       <120>
830 REM WENN SIE DAS PROGRAMM ABSPEI-         <244>
840 REM CHERN WOLLEN, SOLLTEN SIE NOCH        <241>
850 REM DIE ZEILEN 270-290 ANPASSEN.          <027>

© 64'er
```

Listing 1: Demo zum Lichtgriffel. Bitte mit dem Checksummer eingeben.

Schließen wir den »Lightpen« an den Computer an. Schalten Sie diesen dazu aus. Der Light-Pen-Anschluß ist der einzige an den Control-Port des C 64, der nur einmal, und zwar am Port 1, vorhanden ist. Verbinden Sie den Emitter mit GND und den Kollektor mit Light Pen des Port 1. Das war's schon!

Zur Demonstration habe ich das Programm »Demo zum Lichtgriffel« (Listing 1) geschrieben. Es ist den Joystick-Demos sehr ähnlich. Auch hier werden Punkte auf den Bildschirm gemalt. Nur diesmal erscheinen die Punkte dort, wo Sie den Lichtgriffel auf den Bildschirm setzen. Sie können also den Bildschirm als Eingabeeinheit verwenden und direkt auf ihm zeichnen.

Geben Sie das Programm ein und starten Sie es. Als erstes werden Sie gefragt, ob die erscheinenden Punkte jeweils wieder gelöscht werden oder gesetzt bleiben sollen. Antworten Sie mit einem der angegebenen Buchstaben. Nach wenigen Sekunden, in denen der Farbspeicher schwarz initialisiert wird, erscheint irgendwo ein kleines Rechteck auf dem Bildschirm. Nehmen Sie den Fototransistor zur Hand und führen Sie ihn über die Mattscheibe. Der Punkt folgt dem Lichtgriffel über den Bildschirm.

Auch Lichtgriffel und Tastatur stören einander. Wenn Sie unkontrolliert auf den Tasten herumtippen, sehen Sie, daß der Punkt auf dem Bildschirm hin- und herspringt. Ebenso erscheinen an der Cursorposition wahllos Zeichen, wenn sich der Computer im Direktmodus befindet und Sie den Lichtgriffel über den Bildschirm bewegen. Diese Störungen treten beim VC 20 nicht auf; hier lassen sich Tastatur und Lichtgriffel synchron benutzen.

Sollte sich der Punkt auf dem Bildschirm nicht vom Fleck bewegen, so kann das mehrere Gründe haben: Wenn Sie keinen Fehler beim Abschreiben des Listings gemacht haben (sich der VC 20 in der Grundversion befindet) und der Lichtgriffel korrekt angeschlossen ist, kann nur der Bildschirm zu dunkel eingestellt sein. Drehen Sie den Helligkeitsregler etwas weiter auf; das müßte den Lichtgriffel zum Ansprechen bringen.



Eine Verstärker-Schaltung

Funktioniert das Programm immer noch nicht, wird der Anschluß des Lichtgriffels etwas komplizierter. Sie benötigen eine Verstärkerschaltung, die die zu schwachen Signale vom Lichtgriffel für den Computer verstärkt. Eine solche Schaltung gibt Bild 1 wieder. Sie wird durch die im Schaltbild angegebenen Anschlüsse mit dem Computer verbunden und kann sowohl an den C 64 als auch an den VC 20 angeschlossen werden.

Bild 1. Die Schaltung des Lichtgriffelverstärkers

Halbleiter
1 NE 555 DIL
1 BC 238 C
1 TIL 78

Widerstände
1 kΩ
1 10 kΩ
1 Trimmpoti 10 MΩ

Kondensatoren
1 lnF

Sonstiges
etwas Kabel
Control-Port-Stecker

Stückliste 1. Diese Teile benötigen Sie für den Lichtgriffelverstärker

Die Schaltung ist zum Glück recht einfach, sollte jedoch sorgfältig aufgebaut werden, um Kurzschlüsse zu vermeiden. Sie sollten sich deshalb vielleicht von einem Hobby-Elektroniker helfen lassen, wenn Sie sich noch nie mit Elektronik beschäftigt haben. Die Teile, die Sie für den Verstärker benötigen, entnehmen Sie bitte der Stückliste 1. Die folgende Funktionsbeschreibung der Schaltung ist für den fortgeschrittenen Hobby-Elektroniker gedacht und kann übersprungen werden.

Zur Schaltung: Sie besteht im wesentlichen aus einem IC (Integrated Circuit = Integrierter Schaltkreis) des Typs NE 555. Der NE 555 ist in unserer Schaltung als Monoflop (Monostabile Kippstufe) geschaltet. Wenn das Monoflop durch einen negativen Impuls an Pin 2 des ICs angestoßen wird, verbleibt es eine kurze Zeit in einem Arbeitszustand und fällt dann in einen Grundzustand zurück. Die Triggerimpulse (Pin 2 heißt Trigger) kommen von unserem an den Bildschirm gehaltenen Lichtgriffel. Er sendet einen Impuls, wenn er vom Elektronenstrahl der Bildröhre überstrichen wird. An Pin 3 (das ist der Ausgang des ICs) entstehen so die Impulse, die, vom Transistor verstärkt, dem Light-Pen-Anschluß des Computers zugeführt werden.

Mit Hilfe dieser Verstärkerschaltung funktioniert der Lichtgriffel auch bei einem wesentlich dunkleren Bildschirm. Der optimale Arbeitspunkt der Schaltung wird mit dem 1-Mega-Ohm-Potentiometer eingestellt. Das ist beim C 64 besonders sorgfältig durchzuführen, weil sich sonst wieder einmal Probleme mit den normalen Computerfunktionen ergeben. Drehen Sie einfach solange langsam am Potentiometer, bis der Lichtgriffel ordnungsgemäß funktioniert. Am VC 20 arbeitet die Schaltung problemlos, jedoch muß auch hier der Arbeitspunkt eingestellt werden. Der 1-Kilo-Ohm-Widerstand ist ein Vorwiderstand zum Transistor. Die restlichen Bauteile stellen die Zeit ein, während der das Monoflop im Arbeitszustand verweilt, bis es wieder in den Grundzustand zurückfällt.

Sehen wir uns jetzt das Programm einmal genauer an. In Zeile 390/400 wird der Farbspeicher initialisiert, da die Punkte auf dem Bildschirm schwarz erscheinen sollen. Die Lichtgriffelposition wird vom Computer in X- und Y-Richtung jeweils mit einer Genauigkeit von neun Bit ermittelt. Die acht höherwertigen der neun Bits des X-Wertes stehen in der Speicherstelle 53267 (VC 20: 36870), die des Y-Wertes wieder in der darauffolgenden, 53268 (VC 20: 36871). Das Bit 0 des X/Y-Wertes kann nicht abgefragt werden, wodurch eine Auflösung von nur zwei Bildpunkten zustande kommt. Das ist jedoch für »niederauflösende« Grafik völlig ausreichend, geschweige denn für Steuerfunktionen. In den Zeilen 440 bis 470 werden diese beiden Speicherzellen je fünfmal ausgelesen. Die erhaltenen Werte werden addiert und anschließend durch 5 geteilt (Zeile 480). Diese Mittelwertbildung dient zur Stabilisierung der Werte, denn der X-Wert schwankt ziemlich stark, während die Y-Koordinate recht konstant ist. Das ist seltsamerweise bei beiden Computern so, scheint also systembedingt zu sein.

In den Zeilen 510/520 werden die ausgelesenen Werte in Bildschirmkoordinaten umgewandelt. Wenn man einen Lichtgriffel in eigenen Programmen einsetzen will, muß man hier etwas experimentieren. Jeder muß für jedes Programm neu herausfinden, welche Werte die Koordinaten annehmen sollen (können) und wie er die Werte für dieses Programm am besten aufbereitet. Ab Zeile 540 folgt dann die durch die Programme der letzten Folge schon fast zum Standard gewordene Wertekontrolle, die bei diesem Programm wegen des schwankenden X-Wertes besonders wichtig ist. Die verbleibenden Zeilen dürften hinreichend bekannt sein.

Listing 2 zeigt ein weiteres Demonstrationsprogramm zum Lichtgriffel. Hier wird der Lichtgriffel nicht zum Zeichnen, sondern zum Auswählen von Menüpunkten benutzt. Wegen der größeren Genauigkeit des Y-Wertes wird dazu nur dieser verwendet. Es wird also nur die vertikale Richtung des Lichtgriffels ausgewertet. Sie können mit dem Programm einen der drei Tongeneratoren des C 64/VC 20 bestimmen, auf dem dann ein Ton erklingt; die Tonhöhe hängt vom ausgewählten Tongenerator ab. Den gewünschten Tongenerator bestimmen Sie, indem Sie das entsprechende Feld auf dem Bildschirm mit dem Fototransistor berühren.

In Zeile 470 wird das Lichtgriffel-Y-Register ausgelesen. Aufgrund des erhaltenen Wertes werden dann ab Zeile 490 der angewählte Tongenerator und die Tonhöhe bestimmt. Im C 64-Programm wird dabei nur auf eine direkte Berührung des angegebenen Kästchens in vertikaler Richtung reagiert, während beim VC 20 aus Platzgründen in den Programmzeilen größere Bereiche akzeptiert werden. Wird kein neuer Tongenerator angewählt, ist also T2 (=neuer Tongenerator) gleich T1 (=alter Tongenerator), wird durch Zeile 550 verhindert, daß die Generatoren neu gesetzt werden, weil dadurch eine Tonstörung entsteht.



Wenn Sie den Lichtgriffel öfter in selbstgeschriebenen Programmen einsetzen wollen, sollten Sie sich mit dem Fototransistor einen echten Licht-»Griffel« basteln. Nehmen Sie dazu einen Kugel- oder Filzschreiber und entfernen Sie alle »Innereien«. Löten Sie an die beiden Anschlüsse des Fototransistors zwei ausreichend lange Kabel an und isolieren Sie sie sorgfältig gegeneinander. Ein Kurzschluß kann zwar keinen Schaden anrichten, doch setzt er den Fototransistor außer Betrieb. Stecken Sie den Fototransistor jetzt von hinten in den Stift und befestigen Sie ihn vor der Minenöffnung. Je kleiner diese ist, desto konstanter werden Ihre Werte anschließend sein, weil ein kleinerer Bildschirmausschnitt auf den Fototransistor einwirkt. Sie müssen jedoch aufpassen, daß nicht zu wenig Licht auf den Fototransistor fällt, da er sonst nicht mehr anspricht. In diesem Fall bohren Sie die Minenöffnung einfach etwas auf. Sie können auch mit dem Helligkeitsregler am Fernseher oder Monitor etwas korrigieren. Nach dem Anschluß an den Commputer besitzen Sie dann einen richtigen Lichtgriffel.

Damit ist die Besprechung der Control-Ports vorerst abgeschlossen.

(Tobias Nicol/ev)

```
100 REM *****************************         <238>
110 REM *                           *         <159>
120 REM *   T O N G E B E R   M I T *         <002>
130 REM *   ----------------------- *         <237>
140 REM *                           *         <189>
150 REM *   L I C H T G R I F F E L *         <072>
160 REM *   ----------------------- *         <011>
170 REM *                           *         <219>
180 REM *  BY    TOBIAS       NICOL *         <229>
190 REM *                           *         <239>
200 REM *        NEUWIESENSTRASSE 20 *        <202>
210 REM *                           *         <003>
220 REM *       6000   FRANKFURT  71 *        <042>
230 REM *                           *         <023>
240 REM *****************************         <122>
250 :                                         <226>
260 REM ****** BILDSCHIRMAUFBAU  ******       <126>
270 POKE 53281,1 : POKE 53280,3               <029>
280 PRINT "{CLR,DOWN,2SPACE}BITTE WAEHLEN
    SIE!"                                     <087>
290 PRINT "{2SPACE}--------------------{DOWN
    }"                                        <095>
300 FOR I = 1 TO 3                            <145>
310 PRINT "{2SPACE}A*S"                       <142>
320 PRINT "{2SPACE}B B TONGENERATOR";I        <092>
330 PRINT "{2SPACE}Z*X{2DOWN}"                <001>
340 NEXT I                                    <170>
350 :                                         <072>
360 REM ***** SID-CHIP EINSTELLEN *****       <176>
370 T1 = 4 : SI = 54272                       <238>
380 FOR I = 1 TO 10                           <016>
390 READ A,B                                  <186>
400 POKE SI+A,B                               <027>
410 NEXT I                                    <240>
420 DATA 24,15, 0,207, 1,50, 7,207            <214>
430 DATA 8,100, 14,207, 15,150, 6,240         <138>
440 DATA 13,240, 20,240                       <173>
450 :                                         <172>
460 REM ** LIGHTPENPOSITION ABFRAGEN **       <095>
470 Y = PEEK (53268)                          <176>
480 :                                         <202>
490 REM ***** POSITION AUSWERTEN *****        <183>
500 IF Y >  79 AND Y <  91 THEN T2 =  4       <094>
510 IF Y > 119 AND Y < 131 THEN T2 = 11       <041>
520 IF Y > 159 AND Y < 171 THEN T2 = 18       <174>
530 :                                         <254>
540 REM **** TONGENERATOR TAUSCHEN ****       <025>
550 IF T2 = T1 THEN 460                       <136>
560 POKE SI+T1,0                              <078>
570 POKE SI+T2,17                             <183>
580 :                                         <048>
590 REM ** NEUE WERTE & RUECKSPRUNG **        <024>
600 T1=T2                                     <119>
610 GOTO 460                                  <150>
620 :                                         <088>
630 :                                         <098>
640 :                                         <108>
650 REM ******************************        <024>
660 REM *** AENDERUNGEN FUER VC-20 ***        <233>
670 REM ******************************        <044>
680 REM                                       <234>
690 REM LOESCHEN SIE 370-440!                 <209>
700 REM                                       <254>
710 REM GEBEN SIE EIN:                        <101>
720 REM                                       <018>
730 REM 270 POKE 36879,27                     <160>
740 REM 370 T2 = 36874 : POKE 36878,15        <153>
750 REM 470 Y = PEEK (36871)                  <085>
760 REM 500 IFY< 63THENT2=36874:GOTO540       <101>
770 REM 510 IFY< 85THENT2=36875:GOTO540       <025>
780 REM 520 IFY<103THENT2=36876:GOTO540       <109>
790 REM 560 POKE T1,0                         <158>
800 REM 570 POKE T2,(T2-36871)*50             <060>
```

© 64'er

Listing 2. Tongeber mit Lichtgriffel. Bitte mit dem Checksummer eingeben.

Sortieren mit dem Computer (5)

Ein ausführlicher Geschwindigkeitsvergleich aller bisher erarbeiteten Sortieralgorithmen soll die Spreu vom Weizen trennen. Und ein weiteres Sortieralgorithmus wird vorgestellt.

Entgegen der anderslautenden Vorhersage im letzten Teil unseres Kurses, soll heute noch einmal auf Sortieralgorithmen in Basic eingegangen werden. Dieser Entschluß wurde aus mehreren Gründen gefaßt.

Erstens trafen nach Erscheinen der letzten Ausgabe mehrere interessante Beiträge zum Thema Sortieren in der Redaktion ein. Zweitens soll noch einmal ein grafischer Vergleich sämtlicher Routinen stattfinden, und drittens werden wir uns heute noch ein wenig mit den Vor- und Nachteilen der einzelnen Sortiermethoden beschäftigen.

Als erstes soll eine Verbesserung an den Mann gebracht werden. Heinrich Studer schickte der Redaktion einen Sortieralgorithmus, der dem schon bekannten Straight Selection stark ähnelt, wegen seines einfacheren Aufbaus jedoch weniger Speicherplatz benötigt. Der Einfachheit halber soll dieser Sortieralgorithmus den Namen Minisort bekommen.

Wie Sie aus dem Flußdiagramm in Bild 1 sehen können, besteht Minisort aus zwei geschachtelten FOR-NEXT-Schleifen. In der äußeren Schleife wird jeweils der Reihe nach ein Element genommen und mit sämtlichen Elementen der inneren Schleife verglichen. Wird während eines Vergleichs festgestellt, daß das andere Element aus dem Restfeld kleiner als unser Vergleichswert ist, so werden beide Variablen vertauscht.

Ist die innere Schleife also einmal durchlaufen worden, so befindet sich das kleinste Element des Gesamtfeldes nun an erster Stelle.

Jetzt wird in der äußeren Schleife zum nächsten Element übergegangen, wobei dieses wiederum mit dem Restfeld verglichen wird... und so weiter, bis man beim letzten Element angekommen ist. Es muß sich hierbei dann zwangsläufig um den größten Wert handeln.

Für Minisort (Listing 1) wie auch für unsere übrigen Sortieralgorithmen gilt, daß sie das Hauptprogramm für ihren Betrieb benötigen. Das Hauptprogramm dient der Erstellung der verschiedenen Arrays und der Anzeige der erstellten und sortierten Felder. In Listing 2 wurde auch diesmal wieder das Hauptprogramm abgedruckt, da es um ein paar Sachen erweitert wurde. Es ist jetzt mit der Option »von Hand erstellen« zusätzlich möglich, jeweils vorwärts und rückwärts sortierte Felder zu erstellen, um die »Testkandidaten« (sprich: Sortierprogramme) auf Herz und Nieren zu testen.

Um Ihnen jedoch einen großen Teil der Arbeit abzunehmen, habe ich für diese Folge unseres Kurses noch einmal sämtliche Sortierprogramme, die bisher besprochen wurden, in den drei Bereichen »Feld un-

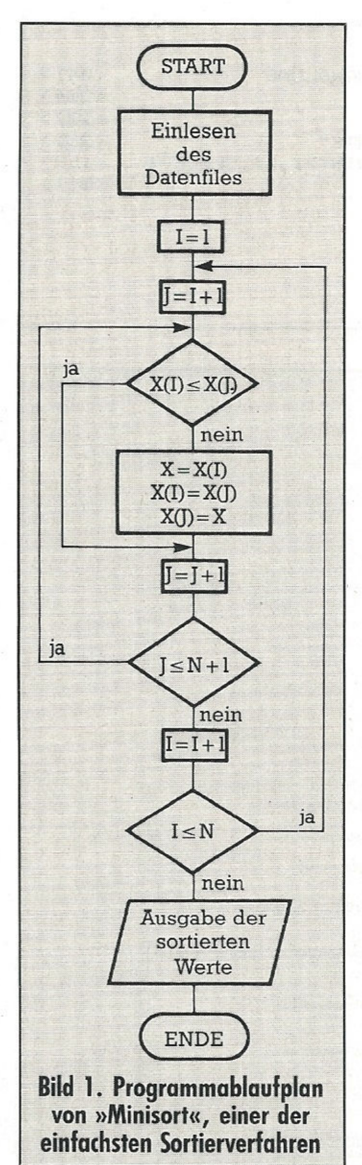


Bild 1. Programmablaufplan von »Minisort«, einer der einfachsten Sortierverfahren

sortiert«, »Feld vorwärts sortiert« und »Feld rückwärts sortiert« untersucht. Wir werden nachher noch ausführlich auf die Ergebnisse dieses »Monstertests« eingehen.

Jetzt jedoch noch zu einem anderen Sortierprogramm, das uns Horst Armin Kosog eingesandt hat. Es trägt den Namen »Mischsort« und wurde von besagtem Absender anhand einer Aufgabenstellung aus einem Informatikbuch (H. Balzert, Informatik 1, Hueber-Holzmann Verlag, München 1976, Seite 210) erstellt und selbständig weiterentwickelt.

Das Prinzip dieses Sortieralgorithmus ist folgendes:

Ähnlich wie bei Quicksort wird das zu sortierende Array bei Mischsort in Untereinheiten (Teilfelder) untergliedert. Diese Teilfelder fangen bei der Größe 2 an und nehmen dann im Quadrat an Umfang zu (4,8,16,...).

Zuerst werden die Untereinheiten der Größe 2 jeweils für sich sortiert, so daß wir A/2 (A ist die Anzahl der Elemente im Gesamtfeld), sortierte Teilfelder erhalten.

Jeweils zwei benachbarte Teilfelder werden nun »zusammengemischt« und wiederum sortiert.

Setzt man dies durch das gesamte Array fort, so erhalten wir nun A/4 sortierte Teilfelder der Länge 4.

Als nächstes werden jeweils zwei benachbarte Teilfelder der Länge 4 zusammengemischt und sortiert... und so weiter, bis schließlich nur noch ein (Teil) Feld übrigbleibt, nämlich das Gesamtfeld.

```
10040 TI$="000000"                              <113>
10050 FOR X=1 TO A-1                            <127>
10060 FOR Y=X+1 TO A                            <008>
10070 IF A$(X)<=A$(Y)THEN 10090                 <242>
10080 S$=A$(X):A$(X)=A$(Y):A$(Y)=S$             <198>
10090 NEXT Y,X                                  <089>
```

© 64'er

Listing 1. Ein kleines und effektives Sortierprogramm: Minisort

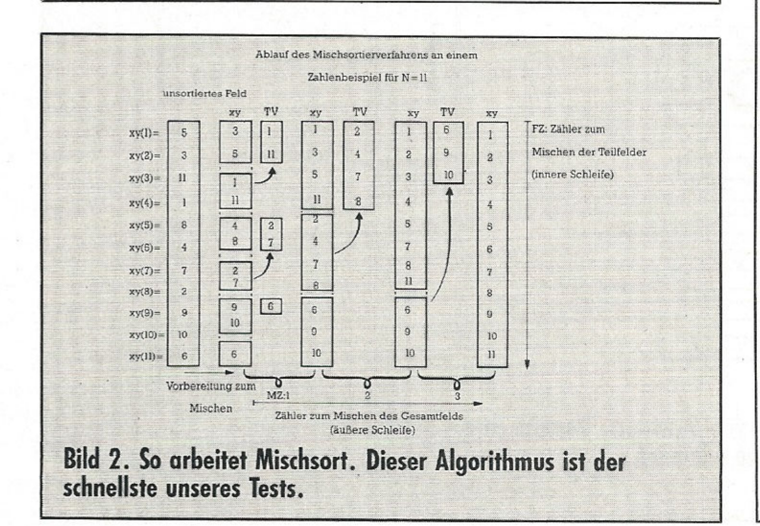


Bild 2. So arbeitet Mischsort. Dieser Algorithmus ist der schnellste unseres Tests.

```
10 REM ERSTELLEN EINES FELDES ZUM               <007>
20 REM SORTIEREN.                               <053>
30 REM DAS ERSTELLEN KANN ZUFAELLIG             <198>
40 REM ODER GEZIEHLT (DURCH EINGABE)            <005>
50 REM ERFOLGEN.                                <251>
60 REM                                          <122>
70 REM SORTIERALGORITHMEN ERHALTEN DIE          <053>
80 REM ZEILENNUMMERN VON 1000 BIS 10000         <167>
90 REM SIE BENOETIGEN JEWEILS DIESEN            <211>
99 REM VORSPANN ZUR AUSFUEHRUNG.                <089>
100 REM HERSTELLUNG EINES ARRAYS:               <090>
110 REM ARRAYVARIABLE         - A$              <220>
120 REM SCHLEIFENVARIABLEN - X, Y, Z            <082>
130 REM HILFSVARIABLEN        - B$, C$, D$      <124>
140 REM DREIECKTAUSCH MIT     - S$              <230>
150 PRINT"{CLR}":CLR                            <102>
160 PRINT"SOLL VON{SPACE,RVSON}H{RVOFF}AND
    ODER{SPACE,RVSON}Z{RVOFF}UFAELLIG ERS
    TELLT":PRINT                                <027>
170 INPUT"WERDEN ";X$                           <239>
180 IF X$<>"H"AND X$<>"Z"THEN 150               <056>
190 IF X$="H"THEN GOSUB 220:GOSUB 1000:GOT
    O 210                                       <161>
200 GOSUB 220:GOSUB 2000                        <223>
210 GOTO 4000: REM WEITERMACHEN                 <132>
220 REM ANZAHL DER ELEMENTE BESTIMMEN           <098>
230 PRINT:INPUT"ANZAHL DER ELEMENTE ";A         <115>
240 IF A>10000 THEN PRINT:PRINT"ZU VIELE E
    LEMENTE":GOTO 230                           <119>
250 IF A<10 THEN PRINT:PRINT"ZU WENIGE ELE
    MENTE":GOTO 230                             <138>
255 DIM A$(A)                                   <078>
260 INPUT"{2DOWN,RVSON}D{RVOFF}RUCKER ODER
    {SPACE,RVSON}B{RVOFF}ILDSCHIRM ";Y$         <207>
270 IF Y$<>"D"AND Y$<>"B"THEN 260               <160>
280 IF Y$="D"THEN D=4:GOTO 300                  <128>
290 D=3                                         <253>
300 RETURN                                      <104>
1000 REM EINGABE VON HAND                       <107>
1010 PRINT"{CLR}V ODER R";                      <093>
1020 INPUT X$:IF X$<>"R"AND X$<>"V"THEN 10
     20                                         <051>
1030 R=1:X1=1:X2=A                              <123>
1040 IF X$="R"THEN R=-1:X1=A:X2=1               <181>
1050 Z1=65:Z2=65:Z3=65:FOR X=X1 TO X2 STEP
     R                                          <127>
1060 A$(X)="":A$(X)=CHR$(Z1):Z1=Z1+1:IF Z1
     >90 THEN Z1=65                             <126>
1062 A$(X)=CHR$(Z2)+A$(X):IF Z1=90 THEN Z2
     =Z2+1:IF Z2>90 THEN Z2=65                  <254>
1064 A$(X)=CHR$(Z3)+A$(X):IF Z2=90 THEN Z3
     =Z3+1:IF Z3>90 THEN Z3=65                  <000>
1070 NEXT X                                     <002>
1080 RETURN                                     <122>
2000 REM ZUFAELLIGE EINGABE                     <177>
2010 PRINT"{CLR}"                               <220>
2020 PRINT:PRINT"ES WERDEN JETZT"A" ELEMEN
     TE ZUFAELLIG":PRINT:PRINT"AUSGEWAEHLT
     "                                          <212>
2030 PRINT:PRINT"JEDES ELEMENT BESTEHT AUS
      3 ZEICHEN.":PRINT:PRINT                   <198>
2040 FOR X=1 TO A                               <232>
2050 A$(X)=""                                   <206>
2060 FOR Y=1 TO 3:A$(X)=A$(X)+CHR$(INT(RND
     (TI)*25)+65):NEXT Y                        <134>
2070 NEXT X                                     <242>
2080 RETURN                                     <037>
3000 REM ZWISCHENAUSGABE DER ELEMENTE           <197>
3010 FOR I=1 TO A-9 STEP 10                     <145>
3020 FOR J=I TO I+9:PRINT#1,A$(J)" ";:NEXT
     J                                          <048>
3030 PRINT#1:NEXT I: PRINT#1                    <051>
3040 RETURN                                     <048>
4000 REM WEITERMACHEN                           <180>
4005 OPEN 1,D                                   <020>
4010 PRINT"{CLR}AUSGABE DES ERSTELLTEN FEL
     DES"                                       <149>
4020 PRINT                                      <056>
4030 GOSUB 3000                                 <192>
4040 REM SORTIERUNG STARTET                     <169>
4050 REM                                        <046>
```

Listing 2. Modifiziertes Hauptprogramm für alle Sortieralgorithmen
(Fortsetzung auf der nächsten Seite)

Diese Arbeitsweise sei noch einmal in Bild 2 grafisch dargestellt.

Damit Sie beim Abtippen von Listings nicht aus der Übung kommen, haben wir auch zu Mischsort ein Programm (Listing 3). Es muß, wie alle übrigen Algorithmen auch, in unser Hauptprogramm eingefügt werden.

Ein dokumentiertes Flußdiagramm zu Mischsort ist in Bild 3 abgedruckt.

Auf unser Hauptprogramm zugeschnitten können Sie Supersort in Listing 4 finden. Es kann ohne weitere Änderungen mit den anderen Routinen dieses Kurses zusammenarbeiten.

Nachdem wir nun eine ziemliche Menge von verschiedenen Sortieralgorithmen kennen, sollen diese jetzt auch einmal im Vergleich betrachtet werden. Für den Anwender stellt sich in der Regel nur die Frage, wel-

```
50000 T$=TI$:REM ENDEBEHANDLUNG              <041>
50010 PRINT#1                                <204>
50020 GOSUB 3000                             <207>
50030 PRINT#1,A;" ELEMENTE"                  <251>
50040 PRINT#1:PRINT#1:PRINT#1,T$:CLOSE 1     <158>
50050 END                                    <011>
```

© 64'er

Listing 2. (Schluß)

Schneller als Quicksort!

Mit dieser Überschrift stellte kürzlich eine Zeitschrift für Personal Computer (Computer persönlich, Ausgabe 14/85) ein neues Sortierverfahren vor. Das Programm lief ursprünglich auf einem CBM 8032 und machte also keine Mühe bei der Umstellung auf den Commodore 64. Ein derartiger Beitrag darf natürlich in unserem Kurs nicht unerwähnt und unbesprochen bleiben, weshalb ich das Programm in unsere Reihe der Sortiermethoden aufgenommen habe.

Das interessante an Supersort (so heißt das Wunderprogramm) ist sicherlich die Art und Weise, wie bei diesem Programm Zeit gespart wird. Wir haben es hier nämlich eigentlich mit zwei Sortierprogrammen zu tun.

Das eine Sortierprogramm davon kennen wir bereits. Es handelt sich um Bubblesort 2, also die verbesserte Version des ursprünglichen Bubblesort. (Wir erinnern uns: die Verbesserung von Bubblesort bestand im Einführen eines Zeigers, der dafür sorgte, daß Bubblesort mit der Arbeit aufhört, sobald keine Vertauschungen mehr stattfinden. Dieser »Kunstgriff« sorgte dafür, daß Bubblesort 2 bei Feldern, die insgesamt schon sortiert sind, einige wenige Elemente aber neu eingeordnet werden müssen, äußerst schnell wird und die kürzesten Sortierzeiten überhaupt erreichen kann.).

Bei Supersort nutzt man nun die Tatsache, daß Bubblesort bei Feldern bis maximal zehn Elementen sehr schnell arbeitet. Es wird also das Gesamtfeld einfach in Teilfelder mit maximal zehn Elementen aufgeteilt, die dann alle einzeln von Bubblesort nachsortiert werden.

Damit die Teilfelder dann auch in der richtigen Reihenfolge stehen, wird das Gesamtfeld vorsortiert, wobei die Teilfelder entsprechend der Anfangsbuchstaben der Feldelemente zusammengestellt werden.

Das Flußdiagramm von Supersort sehen Sie in Bild 4, wobei auf die detaillierte Darstellung von Bubblesort verzichtet wurde.

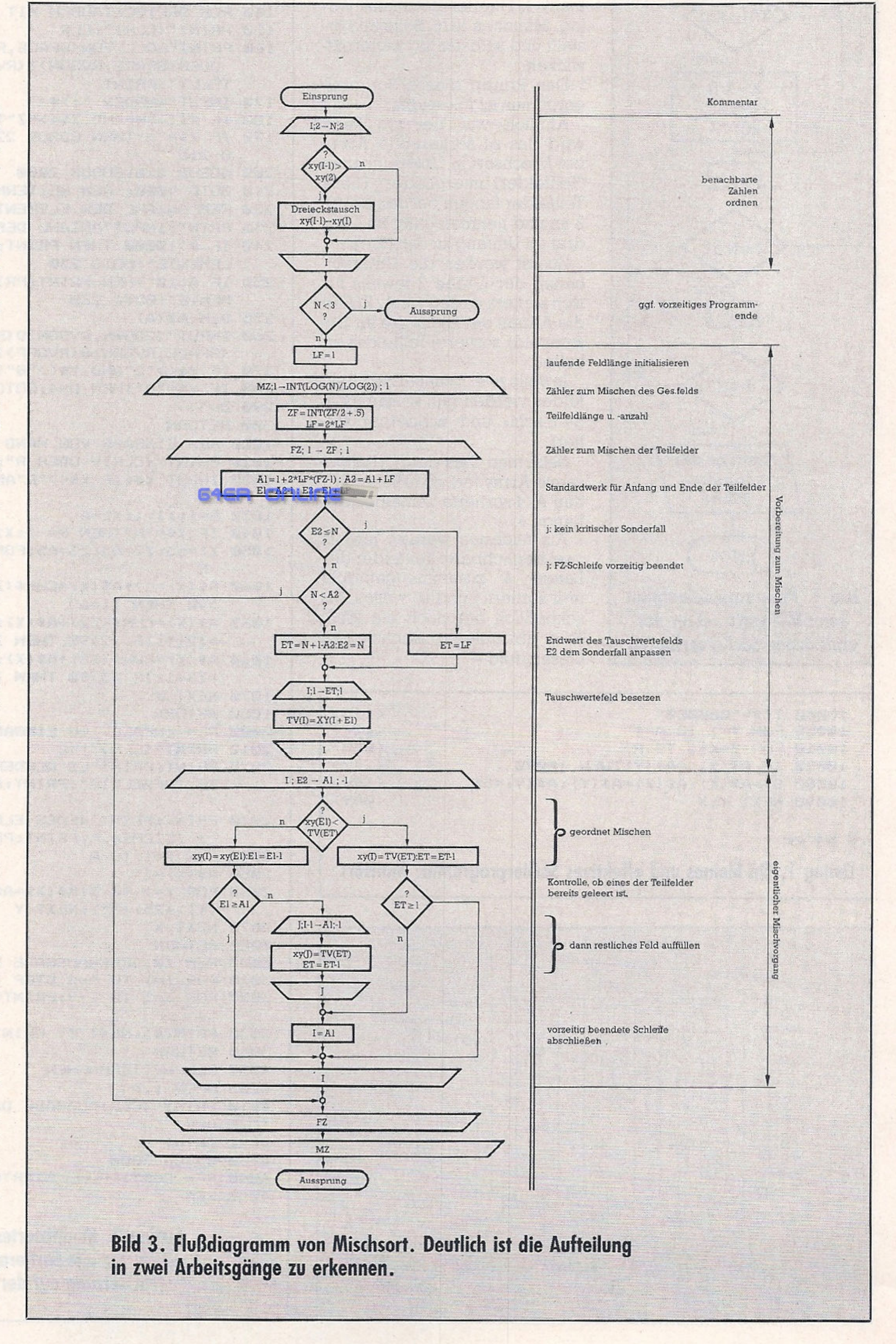


Bild 3. Flußdiagramm von Mischsort. Deutlich ist die Aufteilung in zwei Arbeitsgänge zu erkennen.



```
10040 TI$="000000"                                    <113>
10050 ZF=INT(A/2+.5):DIM TV$(ZF)                      <179>
10060 FOR X=2 TO A STEP 2                             <169>
10070 IF A$(X-1)>A$(X)THEN TV$(0)=A$(X-1):
      A$(X-1)=A$(X):A$(X)=TV$(0)                      <203>
10080 NEXT X:IF A<3 THEN 50000                        <163>
10090 LF=1:FOR MZ=1 TO INT(LOG(A-1)/LOG(2)
      )                                               <009>
10100 ZF=INT(ZF/2+.5):LF=2*LF                         <057>
10110 FOR FZ=1 TO ZF:A1=1+2*LF*(FZ-1):A2=A
      1+LF:E1=A2-1:E2=E1+LF                           <030>
10120 IF E2<=A THEN ET=LF:GOTO 10150                  <009>
10130 IF A<A2 THEN 10230                              <207>
10140 E2=A:ET=A+1-A2                                  <023>
10150 FOR X=1 TO ET:TV$(X)=A$(X+E1):NEXT              <003>
10160 FOR X=E2 TO A1 STEP-1                           <048>
10170 IF A$(E1)<TV$(ET)THEN 10200                     <183>
10180 A$(X)=A$(E1):E1=E1-1:IF E1>=A1 THEN
      10220                                           <036>
10190 FOR Y=X-1 TO A1 STEP-1:A$(Y)=TV$(ET)
      :ET=ET-1:NEXT Y:GOTO 10210                      <075>
10200 A$(X)=TV$(ET):ET=ET-1:IF ET>=1 THEN
      10220                                           <056>
10210 X=A1                                            <096>
10220 NEXT X                                          <006>
10230 NEXT FZ,MZ                                      <042>

© 64'er
```

Listing 3. Das schnellste Sortierprogramm: Mischsort



```
10000 TI$="000000":DIM ZP$(A),AN$(26),X(26
      ),B$(A)                                         <182>
10040 FOR II=1 TO 26                                  <252>
10050 AN$(II)="":X(II)=0                              <173>
10060 NEXT II                                         <130>
10070 IF ASC(A$(1))>64 THEN MIN=64                    <158>
10080 IF ASC(A$(1))>192 THEN MIN=192                  <153>
10085 FOR II=1 TO A                                   <204>
10090 JJ=ASC(A$(II))-MIN                              <239>
10100 IF JJ<0 THEN JJ=26                              <215>
10110 AN$(JJ)=AN$(JJ)+STR$(II+100)                    <073>
10120 NEXT II                                         <190>
10130 L=1                                             <185>
10140 FOR JJ=1 TO 26                                  <120>
10150 X(JJ)=LEN(AN$(JJ))/4                            <128>
10160 IF AN$(JJ)=""THEN 10210                         <018>
10170 FOR II=1 TO LEN(AN$(JJ))STEP 4                  <042>
10180 X=(VAL(MID$(AN$(JJ),II,4))-100)                 <011>
10190 B$(L)=A$(X):A$(X)="":L=L+1                      <194>
10200 NEXT II                                         <014>
10210 NEXT JJ                                         <048>
10220 FOR II=1 TO A                                   <083>
10230 A$(II)=B$(II):B$(II)=""                         <050>
10240 NEXT II                                         <056>
10250 Y=0                                             <087>
10260 FOR L=1 TO 26                                   <149>
10270 X=Y+1:Y=X+X(L)-1                                <105>
10280 IF X(L)=0 OR X(L)=1 THEN 10340                  <033>
10290 FOR JJ=Y-1 TO X STEP-1:FL=-1                    <160>
10300 FOR N=X TO JJ                                   <210>
10310 IF A$(N)>A$(N+1)THEN FL=0:TE$=A$(N):
      A$(N)=A$(N+1):A$(N+1)=TE$                       <133>
10320 NEXT N                                          <028>
10330 IF NOT FL THEN NEXT JJ                          <018>
10340 NEXT L                                          <032>

© 64'er
```

Listing 4. Das Programm Supersort für den C 64

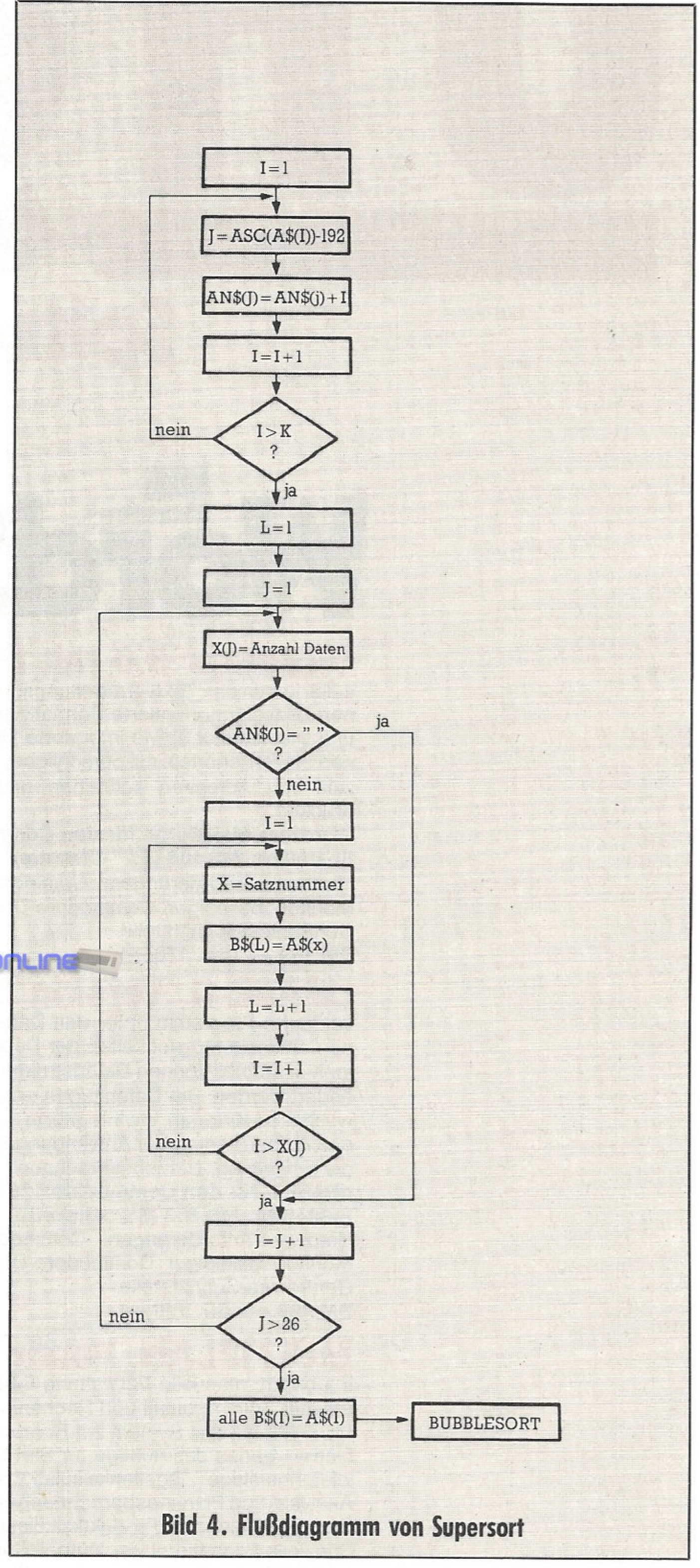


Bild 4. Flußdiagramm von Supersort

ches Sortierprogramm er für seine Problemlösung am besten verwenden kann, und diese Frage ist mitunter nicht so leicht zu beantworten.

Für den Vergleichstest habe ich folgende Sortierprogramme verwendet:

Straight Insertion, Straight Selection, Bubblesort 2, Shellsort, Heapsort, Quicksort, Supersort und Mischsort.

Fangen wir mit den Bedingungen des Tests an. Die einzelnen Programme wurden mit vorsortierten und unsortierten Feldern getestet, wobei nur Stringarrays verwendet wurden. Während des Tests wurde die Anzahl der Elemente jedesmal vergrößert, damit sich anschließend ein Kurvendiagramm darstellen ließ.

Die ersten fünf Algorithmen wurden mit 10, 20, 50, 100 und 200 Elementen gemessen (Bubblesort nur bis 100). Die drei schnelleren Methoden Quicksort, Supersort und Mischsort wurden auch noch mit 500 Elementen geprüft.

Höhere Anzahlen von Feldelementen waren aus mehreren Gründen nicht nötig.

Erstens steigt die Sortierzeit einiger Routinen nach kurzer Zeit enorm an, was zu Werten außerhalb der Grafikskala führt.

Zweitens kommt es zu Konflikten mit der Geduld des Testers, wenn zu viele Elemente verwendet werden (Bubblesort 2 benötigte beispielsweise bei 1000 rückwärts geordneten Elementen sage und schreibe 6702 Sekunden; das sind 1 Stunde 51 Minuten und 42 Sekunden!). Und drittens kommt ab einer gewissen Zahl an Feldelementen noch ein ganz anderes Problem zum Tragen, das sich Garbage Col- ►





lection nennt und auch den schnellsten Sortieralgorithmus zur Verzweiflung bringt.

Nun also zur Besprechung der Testergebnisse:

Sehen Sie sich einmal Bild 5 an, es enthält die Testergebnisse von Bubblesort einmal grafisch dargestellt.

Auf der X-Achse ist dabei die Anzahl der Elemente und auf der Y-Achse die benötigte Sortierzeit in Sekunden angeführt.

Wie man erkennen kann, steigt die Sortierzeit dabei exponentiell an, was auf lange Wartezeiten hoffen läßt. Eine Ausnahme bilden jedoch die schon sortierten Felder. Hier zeigt sich Bubblesort von seiner besten Seite. Es stellt innerhalb von wenigen Sekunden fest, ob ein Feld schon sortiert ist und eignet sich deshalb hervorragend zum Einordnen weniger Elemente in ein großes sortiertes Feld.

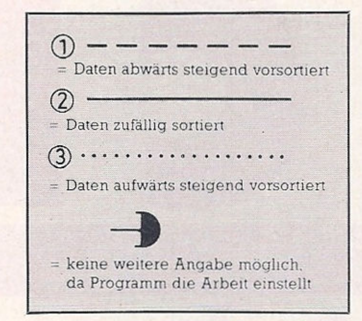
Bild 6 zeigt Straight Insertion in Aktion. Dieses Sortierprogramm zeigt ein ähnliches Verhalten wie Bubblesort und ist sogar noch ein wenig schneller. Die Anwendungsgebiete liegen also auf der gleichen Ebene wie bei Bubblesort: Einordnen weniger Elemente in ein großes vorsortiertes Feld (vielleicht ließe sich durch Straight Insertion Supersort noch ein wenig mehr »aufmöbeln« ?!?).

Bild 7 macht uns mit der Charakteristik von Straight Selection vertraut, und man kann nicht verhehlen, daß dieser Sortieralgorithmus auf allen Gebieten langsam und uneffektiv arbeitet. Er lohnt sich der Einfachheit halber bei sehr kleinen Feldern mit unterschiedlichen Sortierungen.

Bild 8 zeigt uns den ersten effektiven Algorithmus, der durchaus schon praktisch eingesetzt werden kann. Es handelt sich um Shellsort, und wir können unschwer erkennen, daß diese Kurve im Gegensatz zu der von Straight Selection schon ziemlich flach verläuft, was einen langsameren Anstieg der Sortierzeiten bei steigender Anzahl von Elementen zur Folge hat.

Die Kurve in Bild 9 hat einen ganz ähnlichen Verlauf zur Kurve in Bild 8, nur ist die Steigung wieder um ein Stück flacher geworden. Heapsort läßt also durchaus auch praktische Anwendungen zu, es wird jedoch wegen seiner komplizierten Struktur kaum eingesetzt.

Bild 10 zeigt den Zeitverlauf bei Quicksort. Das Erstaunliche an den höheren Sortieralgorithmen ist, daß sich alle Sortierzeiten, sowohl die von völlig unsortierten Feldern als auch die von sortierten Feldern ständig aneinander annähern, was auf einen sehr breit gefächerten Anwendungskreis dieser Sortiermethoden schließen läßt.



Eine Sache sollte im Zusammenhang von Quicksort nicht unerwähnt bleiben: Supersort bedient sich Bubblesort zur Endsortierung seiner Teilfelder. Diese Methode ist auch bei Quicksort üblich, wenngleich wir das in unserem Kurs nicht berücksichtigt haben. Es zeigt sich nämlich in der Praxis, daß die »großen« Sortiermethoden bei kleiner werdenden Teilfeldern mitunter sehr schwerfällig arbeiten, weshalb es besser ist, diese Teilfelder (zirka 10 Elemente) an kleine (und bei diesen Größen auch schnelle) Sortieralgorithmen zu übergeben.

Bild 11 macht uns mit der Geschwindigkeit von Supersort vertraut und bringt schon die erste Überraschung:

Bei unsortierten Feldern bis zu 400 Elementen und bei aufwärts sortierten ist der Sortieralgorithmus schneller als Quicksort, bei abwärts sortierten langsamer. Außerdem zeigte sich Supersort im Test wenig entgegenkommend, was seine Zuverlässigkeit betrifft.

Erstens baut Supersort seine Teilfelder mit Hilfe einer Stringvariable auf, die bekanntermaßen nur 255 Zeichen lang werden kann. Das hat den Haken, daß Supersort mit einer »?STRING TOO LONG«-Meldung ausstieg, sobald eine ganze Reihe von gleichartigen Elementen verarbeitet werden mußten. Das ist jedoch bei einem vorsortierten Feld leider oft der Fall. Aus diesem Grund mußte ich bei einer Zahl ab 100 sortierten Elementen mit dem Test aufhören.

Zweitens fiel bei Supersort unangenehm auf, daß diese Sortierroutine nicht in der Lage ist, Strings zu sortieren, die Zahlen enthalten, was auf die Beschränkung auf 26 Zeichen des Alphabets zurückzuführen ist.

Hätte Supersort, abgesehen von den obigen Schwächen, einwandfrei gearbeitet, so hätte man sicher auf interessante Meßwerte kommen können. Es zeigte sich jedoch schon bei den zufallssortierten Elementen, daß Supersort auf ganz andere Schwierigkeiten stößt, wenn die Zahl der Elemente über 500 hinaus geht. Auch hier wieder das berüchtigte Wort Garbage Col-

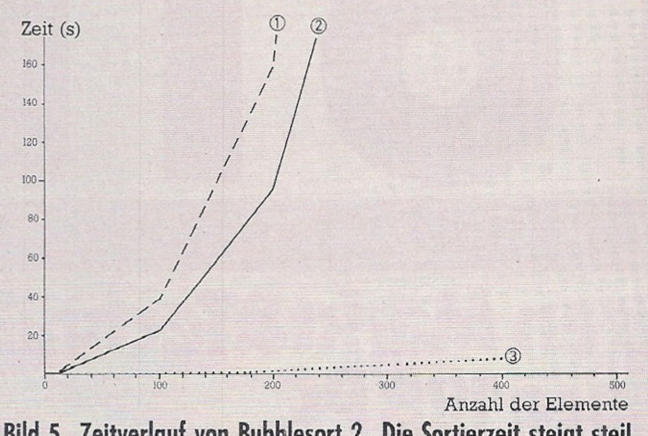


Bild 5. Zeitverlauf von Bubblesort 2. Die Sortierzeit steigt steil exponentiell an.

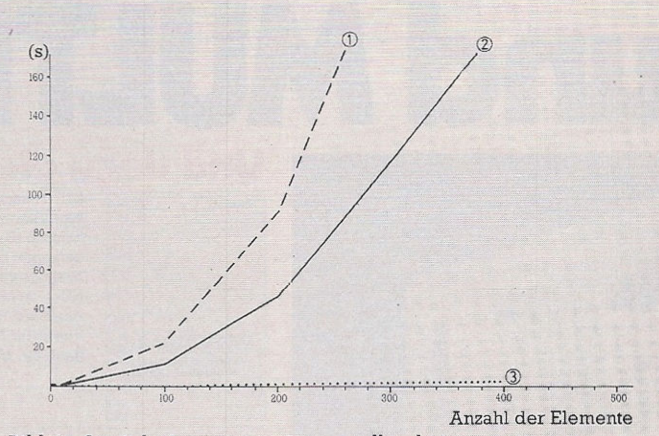


Bild 6. Straight Insertion zeigt vor allen bei vorsortierten Feldern seine Stärke

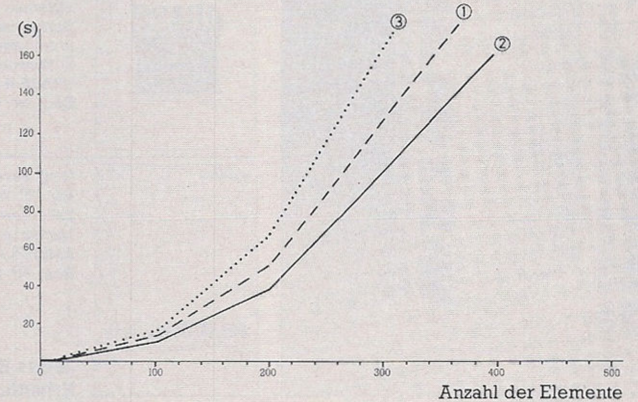


Bild 7. Straight Selection benötigt große Sortierzeiten und findet kaum praktischen Einsatz

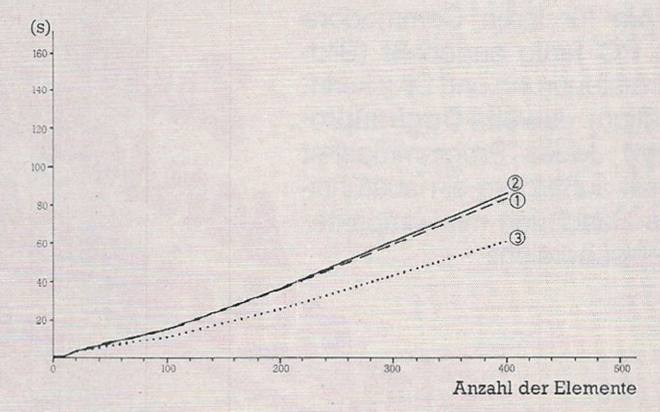


Bild 8. Shellsort ist schon ein »professioneller« Algorithmus

Bild 9. Heapsort zählt zu den schnellsten Sortiermethoden, wie die flache Kurve zeigt

Bild 10. Quicksort war lange der schnellste Sortieralgorithmus

Bild 11. Supersort zeigt einige Mängel auf dem C 64, kann aber spezialisiert hervorragend eingesetzt werden

Bild 12. Mischsort besticht durch seine Schnelligkeit in allen Bereichen

lection. Supersort benötigt nämlich mehr Zwischenspeicher als Quicksort und wird somit von diesem überholt, sobald bei Supersort die Garbage Collection in Aktion tritt (und das tut sie natürlich früher als bei Quicksort).

Natürlich kann man jetzt den Entwickler dieses Programms nicht des Lügens bezichtigen, wenn er behauptet: »Supersort ist schneller als Quicksort!«. Supersort ist nämlich ursprünglich auf einem CBM 8032 geschrieben worden, und dieser verfügt über eine andere Stringverwaltung als der C 64, was eine erheblich schnellere Garbage Collection zur Folge hat

Für den C 64 gilt jedoch (leider): Supersort ist nur bedingt brauchbar, da es einige Schwächen aufweist, die Quicksort nicht hat.

Nun zu Bild 12. Es zeigt den letzten der Sortieralgorithmen, nämlich Mischsort. Und hier erlebte ich die große Überraschung: Auch Mischsort ist schneller als die Normalversion von Quicksort (ohne Tricks mit kleineren Sortierroutinen). Mischsort zeigte im Test wahrhaftig traumhafte Zeiten für Basic-Sortierprogramme, die teilweise nur 50 Prozent der Zeiten von Quicksort betrugen. Es wurde auch deutlich, daß bei Mischsort die Garbage Collection noch später einsetzte als bei Quicksort (das Einsetzen der Garbage Collection ist an den »Knickstellen« der Graphen zu erkennen), was die guten Zeiten aber nur teilweise rechtfertigte.

Mischsort ist genauso einsatzfähig wie Quicksort (ohne »Macken«) und kann deshalb als bestes Sortierprogramm dieses Tests betrachtet werden. Haben Sie viele große Felder zu sortieren, so spielt es bei diesem Algorithmus fast keine Rolle, ob die Felder vorsortiert sind oder nicht; schnell ist er auf jeden Fall.

Noch eine Randbemerkung zum Schluß: Vielleicht haben Sie auch schon Methoden entwickelt, wie sie die Garbage Collection »in den Griff bekommen«. Haben Sie vielleicht als »Maschinensprachefreak« ein neues Basic-ROM geschrieben (mit besserer Stringverwaltung) oder haben Sie anhand unseres Vorkurses über Strings die Lösung des Problems gefunden?

Die 64'er-Redaktion würde sich über Beiträge aus dem Leserkreis freuen. Vielleicht möchten Sie dazu beitragen, daß auch andere Anwender in den Genuß entsprechender Verbesserungen kommen?

Schreiben Sie uns doch! Bestimmt können wir dann den einen oder anderen Trick mit in unsere Reihe aufnehmen.

Bis zum nächsten Mal.

(K.Schramm/gk)

```
6067  LDA #$EA
6069  STA 0315
606C  LDA #$0E
606E  STA D020
6071  CLI
6072  RTS
```

Unser Programm ist komplett. Speichern Sie es bitte vor dem Starten ab. Nach dem SYS 24576 finden Sie einen hübschen bunten Rahmen vor, oberhalb und unterhalb des Textfensters ist er schwarz. Besonders gut — finde ich — sieht das Ganze aus, wenn man die Hintergrundfarbe des Textfensters auch auf Schwarz setzt. Das Programm erlaubt noch einige Experimente:

Durch POKE-Kommandos in die Speicherstelle 2 kann die aktuelle Streifenbreite variiert werden, durch POKEs in die Zelle 24645 der Startwert der Verzögerungsschleife. Probieren Sie's doch mal aus. Eine Erkenntnis werden Sie gewinnen: In der Unterbrechungs-Programmierung spielt die Zeit eine wichtige Rolle. Das zeigt sich auch, wenn man zum Beispiel Cursorbewegungen durchführt: Die Streifen fangen an zu wandern.

Weitere Möglichkeiten zum Experimentieren sind gegeben, wenn Sie die Rasterzeilen verändern, die den oberen und unteren Rand des Textfensters markieren:

Durch POKE 24661,Zahl verschieben Sie die obere, durch POKE 24635,X:POKE 24588,X die untere Rasterzeile, von der an alles schwarz ist. Wie schon vorhin erwähnt, habe ich im Programm diese Werte auf 50 beziehungsweise 248 fixiert, weil genau dort auf meinem Monitor das Textfenster liegt.

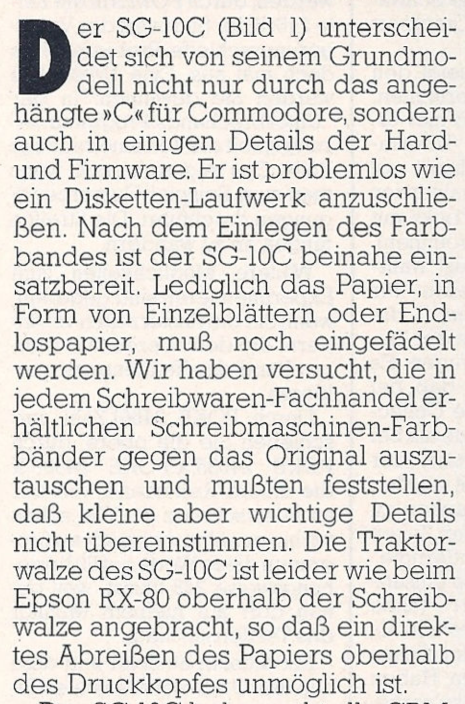
Mit diesem Beispiel und dem aus der Grafikserie sollte es Ihnen nun möglich sein, auch andere Unterbrechungsprogramme zu schreiben, die sich der Rasterzeilen-Unterbrechung per VIC-II-Chip bedienen. Eine Bemerkung sollte ich Ihnen noch auf den Weg Ihrer eigenen Versuche mitgeben: Der Elektronenstrahl, der über den Bildschirm saust und beim Erreichen des von uns bestimmten Rasterzeilenwertes zum Auslösen des IRQ führt, ist enorm schnell. Die Serviceprogramme dürfen deshalb nicht zu lang sein, sonst steht der nächste IRQ schon wieder an, bevor der vorangegangene bearbeitet ist.

In der kommenden Folge sollen Beispiele für andere Unterbrechungsformen vorgestellt werden, die durch die CIAs und die RESTORE-Taste angesprochen werden. Danach soll es um die Anwendung des bisher Gelernten gehen, wobei wir uns wieder mehr den Interpreter-Routinen und auch den Kernal-Möglichkeiten zuwenden wollen.

(Heimo Ponnath/gk)

Star SG-10C — wie hast du dich verändert!

Der SG-10C macht Schluß mit dem Interfaceproblem. Statt der üblichen Centronics-Schnittstelle besitzt er den seriellen IEC-Bus. Unser Test zeigt, welche Vor- und Nachteile damit verbunden sind.



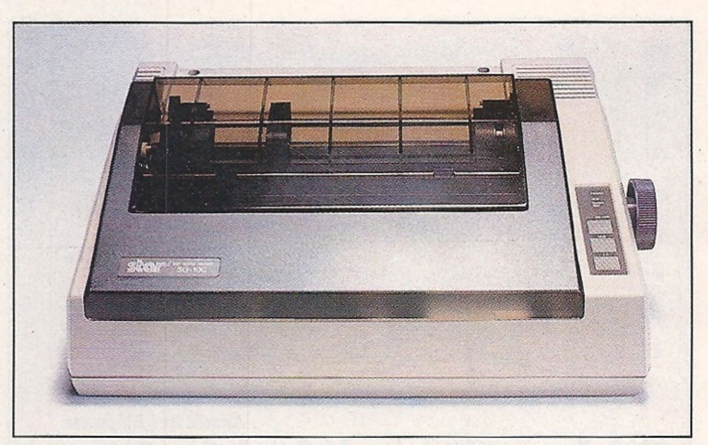


Bild 1. Anschlußfertig an den C 64: der Star SG-10C

Der SG-10C (Bild 1) unterscheidet sich von seinem Grundmodell nicht nur durch das angehängte »C« für Commodore, sondern auch in einigen Details der Hard- und Firmware. Er ist problemlos wie ein Disketten-Laufwerk anzuschließen. Nach dem Einlegen des Farbbandes ist der SG-10C beinahe einsatzbereit. Lediglich das Papier, in Form von Einzelblättern oder Endlospapier, muß noch eingefädelt werden. Wir haben versucht, die in jedem Schreibwaren-Fachhandel erhältlichen Schreibmaschinen-Farbbänder gegen das Original auszutauschen und mußten feststellen, daß kleine aber wichtige Details nicht übereinstimmen. Die Traktorwalze des SG-10C ist leider wie beim Epson RX-80 oberhalb der Schreibwalze angebracht, so daß ein direktes Abreißen des Papiers oberhalb des Druckkopfes unmöglich ist.

Der SG-10C beherrscht alle CBM-spezifischen Funktionen (reverse Zeichen, Breitschrift, Grafikzeichen und formatierten Druck). Daneben kann der SG-10C seine eigentliche Herkunft nicht verschweigen: er ist und bleibt ein Star. Der Weg, der beschritten wurde, um beide »Charaktere« des Druckers unter einen Hut zu bringen, ist neu. Alle Commodore-spezifischen Funktionen werden in üblicher Weise durch Sekundäradressen und den CHR$-Befehl ausgewählt. Die Star-Funktionen wie die NLQ-Schrift, die Fett-, Schmal- und Kursivschrift werden mit Schaltern eingestellt. Für die NLQ-Schrift befindet sich der Schalter auf der Gehäuseoberseite. Alle anderen Funktionen erhält man, indem man die gut erreichbaren DIP-Schalter umlegt. Leider sind die Star-spezifischen Funktionen nicht wie beim Grundmodell durch ESC-Sequenzen programmierbar. Dadurch ist es beinahe unmöglich, diese Schriften in längeren Texten abwechselnd zu benutzen. Es wird jedesmal notwendig, den Druck anzuhalten, den Schalter umzulegen, und dann den Drucker aus- und wieder einzuschalten. Ganz anders bei den Commodore-Funktionen. Programmlistings, Steuer- und Grafikzeichen, reverse Schrift und sogar Hardcopys mit Simons Basic werden problemlos ausgeführt. Ein Wechseln der verschiedenen Zeichensätze (wie beim C 64) ist dabei jederzeit möglich. Die eigentliche Besonderheit dieses neuen Konzepts ist die Mischung aus beiden Funktionsblöcken. Kaum ein anderer Drucker erlaubt es, Commodore-Zeichen im Schönschriftmodus (NLQ) auszudrucken. Gerade diese besonders filigrane Schrift mit einer 17x11 Punktmatrix ist es, die begeistert. Die Schriftprobe in Bild 2 zeigt, wie deutlich und klar diese Schrift ist. Im Business Mode (große und kleine Buchstaben) wurde der Commodore-Zeichensatz abgeändert. Anstelle einiger Grafikzeichen besitzt der SG-10C die deutschen Sonderzeichen (Ä,Ö,Ü,ä,ö,ü,ß).

Mit einer Druckgeschwindigkeit von 120 Zeichen/Sec ist der SG-10C kein langsamer Drucker. Trotzdem sinkt die Druckgeschwindigkeit bei der NLQ-Schrift um weit mehr als die Hälfte ab, denn der Druckkopf muß insgesamt viermal über eine Zeile fahren (zweimal mit Druck und zweimal ohne Druck zurück). In unserem Geschwindigkeitstest erreichte der SG-10C eine sehr gute Zeit von 2:44 Minuten für den Probetext.

Wem alle diese Funktionen nicht reichen, der hat mit dem eingebauten Speicher von 2 KByte Gelegenheit, seine eigenen Zeichen zu definieren. Das umfassende Funktionsangebot wird durch einen Grafikmodus abgerundet. Im Gegensatz zum Grundmodell besitzt der SG-10C aber nur noch eine Punktdichte von 480 Punkten pro Linie.

Nicht für jeden

Der Star SG-10C ist ein Drucker, bei dem die Anpassung an die Commodore-Computer Vor- und Nachteile bringt. Größter Vorteil ist wohl die direkte Anschlußmöglichkeit und die exzellente NLQ-Schrift, auch für CBM-Zeichen. Nachteilig wirkt sich der Verlust der, vom Grundmodell her bekannten, ESC-Befehle zur Schriftenwahl aus. Die Ansteuerung durch DIL-Schalter kann hierfür nur ein bedingt geeigneter Ersatz sein, denn gerade durch den abwechselnden Einsatz verschiedener Schriften läßt sich ein Schriftstück optisch aufmöbeln. Andererseits ist die Programmierung von ellenlangen ESC-Befehlen nicht jedermanns Sache.

Vergleicht man den SG-10C und das Grundmodell (mit externem Star-Interface), so bietet das Grundmodell die eindeutig besseren Leistungen. Deswegen ist der SG-10C mit einem Preis von 998 Mark besonders für diejenigen ein empfehlenswerter Drucker, die einfach drucken möchten, ohne sich viel um Steuerbefehle kümmern zu müssen.

(Arnd Wängler/hm)

```
AUCH DIE COMMODORE-ZEICHEN IN NLQ-QUALITAET
Alle anderen Schriften lassen sich nur mit Schaltern einstellen.
```

Bild 2. Sehr gute Schriftergebnisse liefert der SG-10C

Riteman C+ — der außergewöhnliche Drucker

Drucker können vielerlei Gestalt haben, aber kaum einer sieht so unkonventionell wie der Ritemann C+ aus.

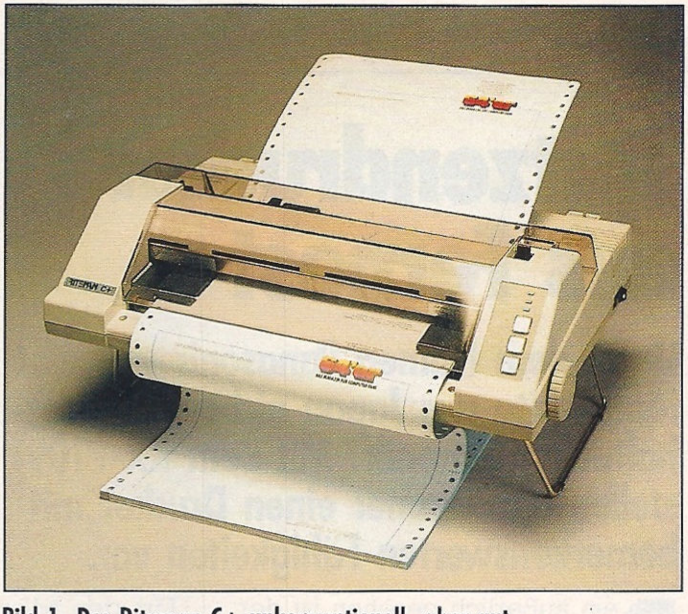


Bild 1. Der Riteman C+ unkonventionell, aber gut

Der Ritemann C+ (Bild 1) durchbricht Normen, angefangen bei der Papierführung, über das Farbband, bis hin zur Art der Aufstellung, ist bei ihm alles anders. Im Gegensatz zu den bisher bekannten Druckergenerationen wird beim Riteman C+ das Papier nicht von hinten, sondern von vorne eingeführt. Ein Traktorband befördert das Papier unter dem Druckkopf durch und schiebt es nach hinten aus dem Drucker. Dieses Konzept schafft, zusammen mit dem ohnehin recht kompakten Drucker, tatsächlich Platz auf dem Computertisch. Nach dem Anstecken am seriellen IEC-Bus des Computers kann es sofort losgehen. Erstes Testkriterium ist die Eignung des Druckers zur Textverarbeitung. Hier wurden wir vom Riteman C+ angenehm überrascht, denn er verfügt über mehrere Funktions-Modi. Im wesentlichen sind das ein Normal- (Commodore) und ein Plus- (EPSON) Modus (Tabelle 1). Die Auswahl des jeweiligen Modus geschieht entweder über zwei der acht DIL-Schalter, die leicht erreichbar unter einer Klappe verborgen sind, oder über einen einfachen ESC-Befehl.

Betrachten wir zunächst den Normal-Modus etwas genauer. Hier wurde versucht, die Fähigkeiten der Commodore-Drucker möglichst vollständig zu implementieren.

Für die Textverarbeitung eignet sich der Normal-Modus aber nur bedingt, denn es stehen keine Umlaute zur Verfügung. Hierfür ist der Plus-Modus prädestiniert, denn er hat die Befehle eines, etwas abgepeckten, Epson FX-80. Im Gegensatz zum Normal-Modus tritt im Plus-Modus die Druckersteuerung durch die ESC-Befehle in den Vordergrund.

Jedem seinen Modus

Der Plus-Modus ist weiterhin mit zwei verschiedenen Grafik-Befehlen (ESC »K« und ESC »L«) ausgestattet. Hier finden Hardcopy-Programme, die für Epson-Drucker geschrieben wurden, Verwendung (8-Nadel-Einzelpunktgrafik). Den eigentlichen Reiz dieser Zweispaltung der Funktions-Modi liegt in der einfachen Umschaltung zwischen Normal- und Plus-Modus. Da dies rein durch Programm-Befehle möglich ist, sind enorme Effekte erzielbar. Einerseits Verwendung der reversen- oder der Grafikzeichen und andererseits alle Funktionen eines EPSON-Druckers sind nur ein Beispiel dafür. Mit einem Textprogramm lassen sich so, mit relativ wenig Aufwand, Rechnungen oder Formulare entwickeln, die neben dem EPSON-Zeichensatz, mit Umlauten, auch den Commodore-Zeichensatz, mit seinen Grafikzeichen, verwenden. Da in beiden Modi alle Zeichen mit vollen Unterlängen gedruckt werden, fällt die Umschaltung kaum auf.

Ein weiteres Testkriterium war natürlich unser Probetext. Der Riteman C+ benötigte hiefür volle 4:20 Minuten. Die Druckgeschwindigkeit ist im Handbuch zwar mit 105 Zeichen/Sekunde angegeben, unser Probetext hat aber gezeigt, daß dieser Wert relativiert werden muß. Von der Bedinungs- und Handhabungsfreundlichkeit des Riteman C+ wird jeder, der mit ihm gearbeitet hat, überzeugt sein. Sowohl das Einlegen des Papiers (Endlos- oder Einzelblattpapier), als auch das Abreißen an der Gehäusekante, sind sehr komfortabel.

Ein gelungener Versuch

Der Riteman C+ ist mit einem Preis von 998 Mark ein außergewöhnlicher Drucker. Auffällig ist das kompakte Design, das ein angenehmes Arbeiten auch auf einem kleinen Tisch erlaubt. Seine beiden Modi gefallen nicht nur allein für sich, sondern auch dann, wenn man sie geschickt miteinander kombiniert. Wer einen professionellen Drucker mit Korrespondenzqualität (NLQ = Near Letter Quality) sucht, ist zwar mit dem etwas teureren Ritemann F+ (1140 Mark) etwas besser bedient. Aber alle, die zu Hause damit arbeiten, und auch auf eine vollständige Commodore-Anpassung nicht verzichten möchten, dürfen das Plus im Namen ruhig für bare Münze nehmen. (aw)

Info: C.Itoh, Roßstr. 96, 4000 Düsseldorf 30, Tel.: 02 11/45 49 80

Plus-(EPSON) Modus	Normal-(Commodore) Modus
senkrechte Formatierung	Commodore Zeichensatz
horizontale Formatierung	Setzen des Zeilenabstandes
horizontale Tabulatoren	Drucker initialisieren
Festlegen der Seitenlänge	Breitschrift
Seitenvorschub	reverser Druck
Breitschrift	Druckposition linker Rand
komprimierte Schrift	Bestimmung der Druckstartposition im Nadelabstand
Fettdruck	7-Nadel Einzelpunktgrafik
Doppeldruck	Festlegen der Seitenlänge
Super-/Subscript	Überspringen d. Perforation
Unterstreichen	Seitenvorschub
Italic (Schrägschrift)	Fixierung der Sekundäradresse
8-Nadel-Einzelpunktgrafik	Zeilenvorschub nicht gekoppelt
einfache Dichte	
doppelte Dichte	Modusänderung (Normal-/Plus-Modus)
Papierendeerkennung an/aus	
Rückwärtsschritt	
unidirektionaler Druck an/aus	
Modus Änderung (Normal-/Plus-Modus)	

```
Der Riteman C+

Dies ist eine Schriftprobe im Plus-Modus, hier stehen alle Umlaute
zur Verfügung:ö Ä ü ö ä ü ß
Obwohl kein NLQ-Modus vorhanden ist, sind die Zeichen sehr schön.
Im Normal-Modus findet man die Commodore-Grafikzeichen:
```

Bild 3. Obwohl keine NLQ-Schrift vorhanden ist, druckt der Riteman C+ sehr sauber

Spitzendrucker aus Fernost

Unter dem Namen Panasonic vertreibt Matsushita aus Japan nicht nur Unterhaltungselektronik. Mit dem KX-P1091 stellen die Japaner einen Drucker mit bemerkenswerten Fähigkeiten vor.

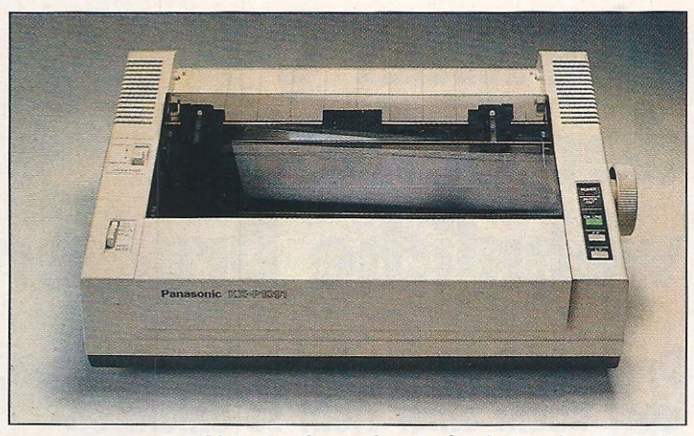


Bild 1. Das kompakte Kraftpaket — der KX-P1091

Rein äußerlich gesehen ist der KX-P1091 (Bild 1) ein recht kompakter Drucker. Nicht aber in seinen Leistungen. Hier haben sich die Konstrukteure einiges einfallen lassen, um ein technisch interessantes Gerät zu entwickeln. Dabei hat man erfreulicherweise daran gedacht, daß es nicht jedermanns Sache ist, einen Drucker einige Minuten lang zu programmieren, nur um eine bestimmte Schrift einzustellen. Das geht natürlich auch. Aber wer es gerne bequem hat, findet auf der vorderen Gehäuseseite einen Schiebeschalter für die wichtigsten Schriftarten. Das Ändern der Schriftart innerhalb einer Zeile ist damit natürlich nicht möglich. Es bleiben dann nur die bekannten ESC-Befehle, die beim Panasonic erfreulicherweise fast 100prozentig mit denen des Epson FX-80 übereinstimmen. Fast jedes Textverarbeitungsprogramm bietet einen Druckertreiber für diesen Pseudo-Standard an. Aber der KX-P1091 kann mehr: Er verfügt über einen NLQ-Modus mit besonders schöner Schrift.

NLQ-Schrift inklusive

Die Schriftprobe in Bild 2 zeigt die Form und Qualität dieser Schrift. Sie verleiht einem Schriftstück eine ansprechende Qualität. Allerdings hat die NLQ-Schrift ihren Preis: Die ursprüngliche Druckgeschwindigkeit von 120 Zeichen/Sekunde verringert sich auf 22 Zeichen/Sekunde.

Für den 64'er-Probetext zur Geschwindigkeitsermittlung brauchte der KX-P1091 2:30 Minuten. Der Epson FX-80 braucht für den gleichen Text 2:15 Minuten. Zusätzlich bietet der KX-P1091 ein großes Maß an Flexibilität. Er kann mit einem CBM-Einbauinterface makellos Listings drucken, ist voll grafikfähig und verarbeitet sowohl Einzel- als auch Endlospapier. Dazu hat der KX-P1091 ähnlich dem Star SR-10 ein Traktorband. Allerdings ist dieses Band oberhalb des Druckkopfes angeordnet, so daß es nicht möglich ist, das Papier direkt über dem Druckkopf abzureißen. Dafür steht der KX-P1091 dem Epson FX-80 in der Grafikfähigkeit fast nicht nach. Mit bis zu vierfacher Punktdichte sind Grafiken in höchster Auflösung für den Panasonic kein Problem und sogar der viel verwendete ESC»*«-Befehl ist vorhanden. Neben diesen wichtigen Befehlen bietet der Panasonic sogar noch eine weitere Funktion zur Beeinflussung des Schriftbildes an: Sein eingebauter Pufferspeicher von 1 KByte erlaubt es eigene Zeichen zu definieren und nach Belieben im Text einzufügen. In der Regel sind die ebenfalls vorhandenen Befehle zur Schriftveränderung wie Fett-, Schmal-, Elite-, Doppeldruck und Breitschrift aber vollkommen ausreichend. Ja sogar einige der selbstdefinierten Zeichen kann man sich sparen. Denn neben einem ASCII-Zeichensatz (Commodore mit Interface) besitzt der Panasonic 64 Blockgrafik und 132 IBM-PC Sonderzeichen (Bild 2). Eine weitere, aber sicherlich nicht unwesentliche, Schriftfunktion ist die Proportionalschrift.

Der Panasonic hat, was so manchem anderen Drucker fehlt. Dazu gehört seine schnell zu wechselnde Farbbandkassette und die leicht zugänglichen acht DIL-Schalter zur dauerhaften Einstellung wichtiger Funktionen. Besonders hervorhebenswert ist das deutsche Handbuch, das beinahe vorbildlich ist.

Schon nach kurzer Zeit hat man den kleinen Schalter auf der Vorderseite des Gehäuses zu schätzen gelernt. Mit einem kurzen Handgriff ist die gewünschte Schrift selektiert. Der Vorteil liegt darin, daß alle anderen Schriftbefehle weiterhin aktiv bleiben, denn es wird lediglich die Grundschrift festgelegt. Unterstreichen, Fettschrift oder Doppeldruck sind (beim NLQ-Modus mit Einschränkungen) — kein Problem.

Der fast ideale Drucker

Der Panasonic KX-P1091 ist ein fast idealer Drucker für den Heimbereich. Durch seine enorme Flexibilität eignet er sich für beinahe jede Aufgabe. Ob Grafik, Listings (mit Interface) oder Briefe — der Panasonic beherrscht es. Gegenüber dem Epson FX-80 kann sich der KX-P1091 durchaus sehen lassen, denn manche Funktion des Epson wird durch den sehr schönen NLQ-Modus mehr als aufgewogen. Das einzige, was an einem Drucker für 1395 Mark (ohne Interface) wie dem KX-P1091, noch zu verbessern wäre, ist der Papierantrieb. Wenn er vor der Schreibwalze angebracht wäre, statt dahinter, ginge beim genauen Positionieren des Papiers nicht jedesmal ein Blatt verloren. Trotzdem, der KX-P1091 ist ein gelungener Wurf.

(Arnd Wängler/hm)

Der PANASONIC KX-1091

Der Panasonic KX-1091 ist ein moderner NLQ (Near Letter Quality) Drucker. Diese NLQ-Zeichen unterscheiden sich vom Normaldruck durch eine wesentlich höhere Auflösung.
Dies ist die Normalschrift (Pica)
Die Elite-Schrift spart Platz
Der Panasonic beherrscht auch die Proportionalschrift.
Die Breitschrift dient
zum hervorheben
Superscript und Subscript
Im Doppeldruck wird alles deutlich
Natürlich mit allen Umlauten: Ö Ä Ü ö ä ü ß
WISSENSCHAFTLICHE SONDERZEICHEN:
Γ π Σ σ μ τ Φ Θ Ω δ ∞ Ø ∈ ∩ ≡ ± ≥ ≤ ⌠ ⌡ ÷ ≈ ° ▪ - √ ⁿ ² ■

Bild 2. Umfangreiche Schriften und eine Schrift in Briefqualität kennzeichnen den KX-P1091

Der Preishammer

Wer bisher glaubte, daß ein gutes Textprogramm wenigstens 99 Mark kostet, wird durch Star Texter positiv überrascht.

Über den Nutzen einer guten Textverarbeitung braucht wohl nicht mehr diskutiert zu werden. Das heißt nicht automatisch, daß alle Textverarbeitungsprogramme nützlich sind. StarTexter, die Textverarbeitung aus dem Sybex-Verlag, ist ein treffliches Gegenbeispiel hierzu. Das Programm wird komplett mit sehr gutem deutschen Handbuch, Diskette und einem Funktionsangebot geliefert, das zu dem Preis von 64 Mark seinesgleichen sucht. Da in Deutschland entwickelt, sind klare deutsche Kommandos und eine an die DIN-Norm angelehnte Tastenbelegung mit den deutschen Sonderzeichen (ö,ä,ü,ß) selbstverständlich. Es ist äußerst angenehm, daß man sich sofort nach dem Laden im Schreibmodus befindet, ohne erst durch ungezählte Menüs hindurch zu müssen. Alle notwendigen Funktionen und Einstellungen werden während der Arbeit mit Star Texter aufgerufen, beziehungsweise vorgenommen. Immer gleichbleibende Parameter, wie der Druckertyp, der Zeichensatz, Tastaturbelegungen, Diskettenlaufwerke und Farbeinstellungen werden, nachdem sie einmal festgelegt wurden, fortan immer automatisch mitgeladen.

Wer schon mit anderen Textprogrammen gearbeitet hat, weiß, daß es Funktionen gibt, die häufig benötigt werden und andere, die zwar wichtig, aber eben seltener gebraucht werden. Wichtige Grundfunktionen sind unter anderem Editieren, Verschieben, Suchen, Ersetzen, Laden und Speichern von Texten (Tabelle 1). Man kann sie auch als Ausschlußkriterien bezeichnen, ohne die alle weiteren Sonderfunktionen ihren Sinn verlieren.

Star Texter verfügt über alle Grundfunktionen. Editieren, Verschieben, Löschen und Kopieren funktionieren schnell und korrekt. Leider sind diese Operationen teilweise blockorientiert, das heißt sie sind immer nur auf ganze Zeilen anwendbar. Während es durchaus möglich ist, beliebig viele Worte mitten in einen Text einzufügen, können Textteile immer nur als ganze Zeilenblöcke verschoben werden.

Neben der Qualität der Grundfunktionen gewinnen die Sonderfunktionen, die für manche Anwendung besonders wichtig sind, an Bedeutung. Gerade in diesem Punkt hebt sich Star Texter von seiner Konkurrenz ab. Zunächst fällt auf, daß es zu keinem Zeitpunkt notwendig ist, die Systemdiskette nochmals einzulegen, dem Anwender bleibt es erspart, sich als Diskjockey zu betätigen.

Wie fast alle bekannten Textprogramme versucht auch Star Texter, die 40-Zeichendarstellung zu umgehen. Das Schreibfeld besteht aus 21 Zeilen und bis zu 80 Spalten. Der Bildschirm wird dabei horizontal verschoben. Ein kleines Fenster am oberen Bildschirmrand gibt jederzeit Aufschluß über die aktuelle Cursorposition in einer Zeile (Bild 1). Andere Fenster informieren über die aktuelle Zeile und Spalte sowie über den jeweiligen Befehlsmodus. Da der Text nach der 21. Zeile ebenfalls vertikal verschoben wird, bis die maximale Zeilennummer 250 erreicht ist, kann man auch von einem großen Arbeitsblatt sprechen. Dieses Arbeitsblatt besteht aus 20000 Zeichen oder umgerechnet etwa sechs bis sieben Schreibmaschinenseiten. Wer mit diesem Textspeicher nicht auskommt, ist auf das Aneinanderbinden einzelner Schriftstücke angewiesen. Erfreulich ist in diesem Zusammenhang, daß Star Texter auch bei fast gefülltem Textspeicher kaum langsamer wird.

Bildschirm = Ausdruck

Manchmal ist es von Nutzen, ein Schriftstück so betrachten zu können, wie es später auf dem Papier abgedruckt wird. Nur so läßt sich feststellen, ob die Absätze stimmen, die Überschriften korrekt sind und der gesamte Text harmonisch verteilt ist. Star Texter bietet dafür einen sehr sinnvollen, weil einfach zu bedienenden, 80-Zeichenmodus an. Durch einfaches Drücken auf die CBM und SHIFT-Tasten wird der gesamte Text umformatiert und kann zwar mit deutlich kleineren Zeichen, aber dafür in seiner Originalbreite, gelesen werden. Drucksonderfunktionen, wie beispielsweise Fettschrift und Unterstreichungen, bleiben (auf dem Bildschirm) allerdings unberücksichtigt. Der 80-Zeichenmodus ist vor allem dann von großem Nutzen, wenn es um die Erstellung von Tabellen und formatierten Texten geht. Für diese Aufgabe bietet Star Texter noch eine weitere

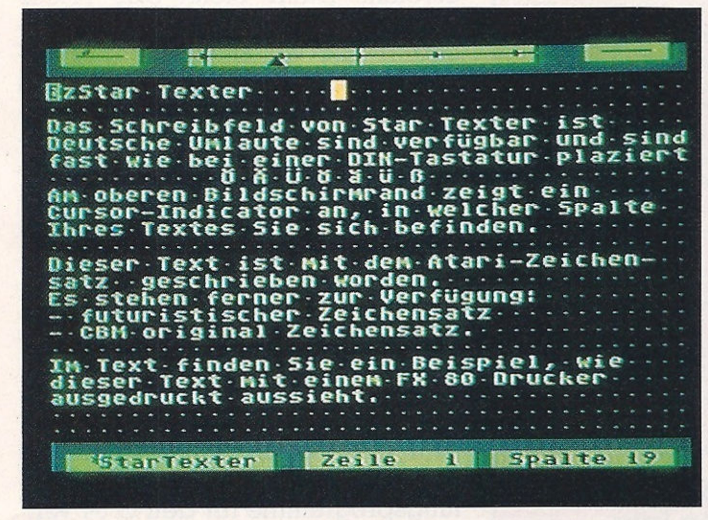


Bild 1. Die Bildschirmbreite wird voll genutzt

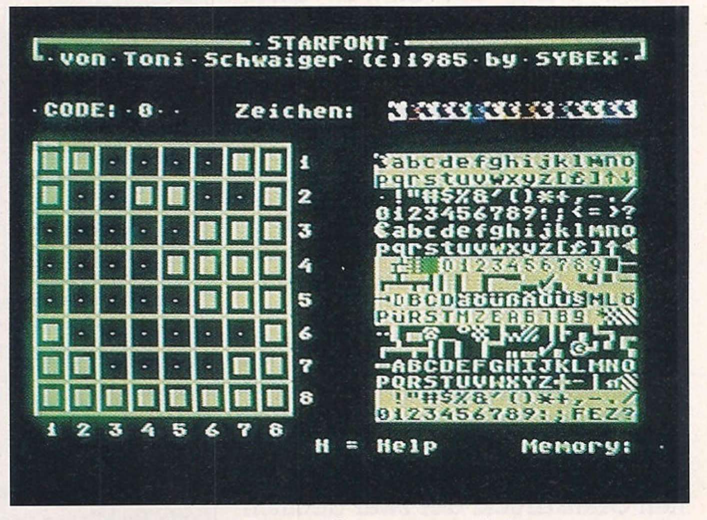


Bild 2. Mit Starfont eigenen Zeichensatz entwickelt

sehr sinnvolle Ergänzung an: die Rechenfunktion. Es sind alle Rechenoperationen und die im Commodore-Basic üblichen Operationen und Befehle gestattet. Der eigentliche Clou an diesem Modus ist seine Programmierbarkeit. Mit einem Befehlsspeicher von 3325 Bytes lassen sich schon manche nützlichen Routinen programmieren und im Text einbinden. So werden Anwendungen denkbar, bei denen im Text verschiedene Werte errechnet und anschließend weiterverarbeitet werden. Das Schreiben von Rechnungen, Listen und Karteikarten könnten solche Anwendungen sein.

Bei der Frage, wie ein bedienungsfreundliches Programm aussehen soll, scheiden sich die Geister. Die beim Star Texter getroffene Menüstruktur stellt eine Mischung aus Baum- und seitenorientierten Menüs dar. Man könnte dieses Prinzip auch als Seitenkonzept bezeichnen, denn Star Texter verwendet nur drei Haupt- und ein kleines Untermenü. Alle Hauptmenüs werden mit den Funktionstasten aufgerufen und sind nach Themenkreisen geordnet. Es gibt ein Diskettenmenü, ein Druckermenü und ein Sammelmenü, das selbst wiederum aus drei Seiten besteht. Während das Floppy- und Druckermenü ihre Funktion selbst erklären, sind im Sammelmenü alle nicht direkt zuzuordnenden Funktionen vereinigt. Einige dieser insgesamt 30 Einstellparameter werden von kaum einem bisher bekannten Programm angeboten. Beispielsweise der Unterpunkt Zeichensatz: Auf der Programmdiskette werden drei vorprogrammierte Zeichensätze, (CBM, Atari, Futura) mitgeliefert. Diese Zeichensätze kann man als Beispiele ansehen, denn prinzipiell ist Star Texter in der Lage, jedes beliebige Zeichen abzubilden. Damit der Entwurf eines neuen Zeichensatzes nicht zum Problem wird, hat man auf der Diskette gleich auch ein Hilfsprogramm abgelegt. Es nennt sich Star Font (Bild 2), ist sehr komfortabel und läßt fast jede Manipulation des Zeichensatzes zu. Die Palette der Anwendungen dieses Modus reicht von einzelnen wissenschaftlichen Zeichen bis hin zum kompletten Zeichensatz. Das ganze Entwerfen eines Zeichensatzes könnte natürlich als netter Gag bezeichnet werden, wenn es da nicht noch eine andere Funktion gäbe, mit der die neuen Zeichen auch auf dem Drucker verewigt werden könnten. Star Texter verfügt über einen Grafikdruck, der zwar deutlich langsamer, aber dafür eben jedes beliebige Zeichen drucken kann. Allerdings wird der Wert der Funktion dadurch geschmälert, daß selbst auf einem Epson- oder Star-Drucker alle Zeichen (auch die nicht geänderten) im »Zitter-Rumpel-Verfahren«, wie es von der MPS 802-Grafik her bekannt ist, ausgegeben werden. Hier wäre es sinnvoll, die Grafikfähigkeit eines angeschlossenen Matrixdruckers besser auszunutzen. Daß der Autor sich über die Fähigkeiten verschiedener Drucker durchaus im klaren war, zeigen zwei andere Funktionen. Zum einen kann die Punktdichte beim Epson mit ESC »*« des Grafikausdrucks variiert werden, zum anderen ist die ESC »!« (Wahl der Schriftart) Funktion durch einen einfachen Parameter einzustellen.

Formatieren leicht gemacht

Es hieße, einiges der Fähigkeiten einer computergestützten Textverarbeitung zu verschenken, wenn man auf einen Formatier-Modus verzichtet. Star Texter tut dies nicht. Er bietet, angewählt durch den Parameter »Trennungen« im Sammelmenü und aufgerufen aus dem Druckermenü, eine sinnvolle Funktion zum Trennen zu langer Worte und Sätze. Setzt man den entsprechenden Parameter, so fragt Star Texter beim Bildschirm-Formatieren (und bei der Komplett-Formatierung) in (fast) jeder Zeile nach, ob ein Wort getrennt werden soll oder nicht. Insbesonders in Verbindung mit dem ebenfalls leicht einstellbaren Blocksatz (rechter und linker Randausgleich) wird so vermieden, daß zu große Lücken in einer Textzeile entstehen. Ebenso einfach wie das Formatieren eines Textes ist auch die Wahl des Druckertyps. Dabei sind für die wohl derzeit am meisten verbreiteten Drucker (Commodore, Epson, Star) spezielle Druckmodi voreingestellt, wobei unterschieden wird, ob ein Drucker mit Harware-Interface (Görlitz, Data Becker, Wiesemann) oder aber nur mit einem Kabel am User-Port angeschlossen ist. Betreibt man einen Drucker mit Centronics-Schnittstelle, kann jegliches Interface (bis auf das Kabel) entfallen, denn Star Texter hat einen eigenen Druckertreiber. Bei solcher Leistungsfähigkeit verwundert es kaum noch, daß alle selbstdefinierten Zeichen und die deutschen Sonderzeichen auch auf den Commodore-Druckern, die dazu normalerweise nicht in der Lage sind, wiedergegeben werden.

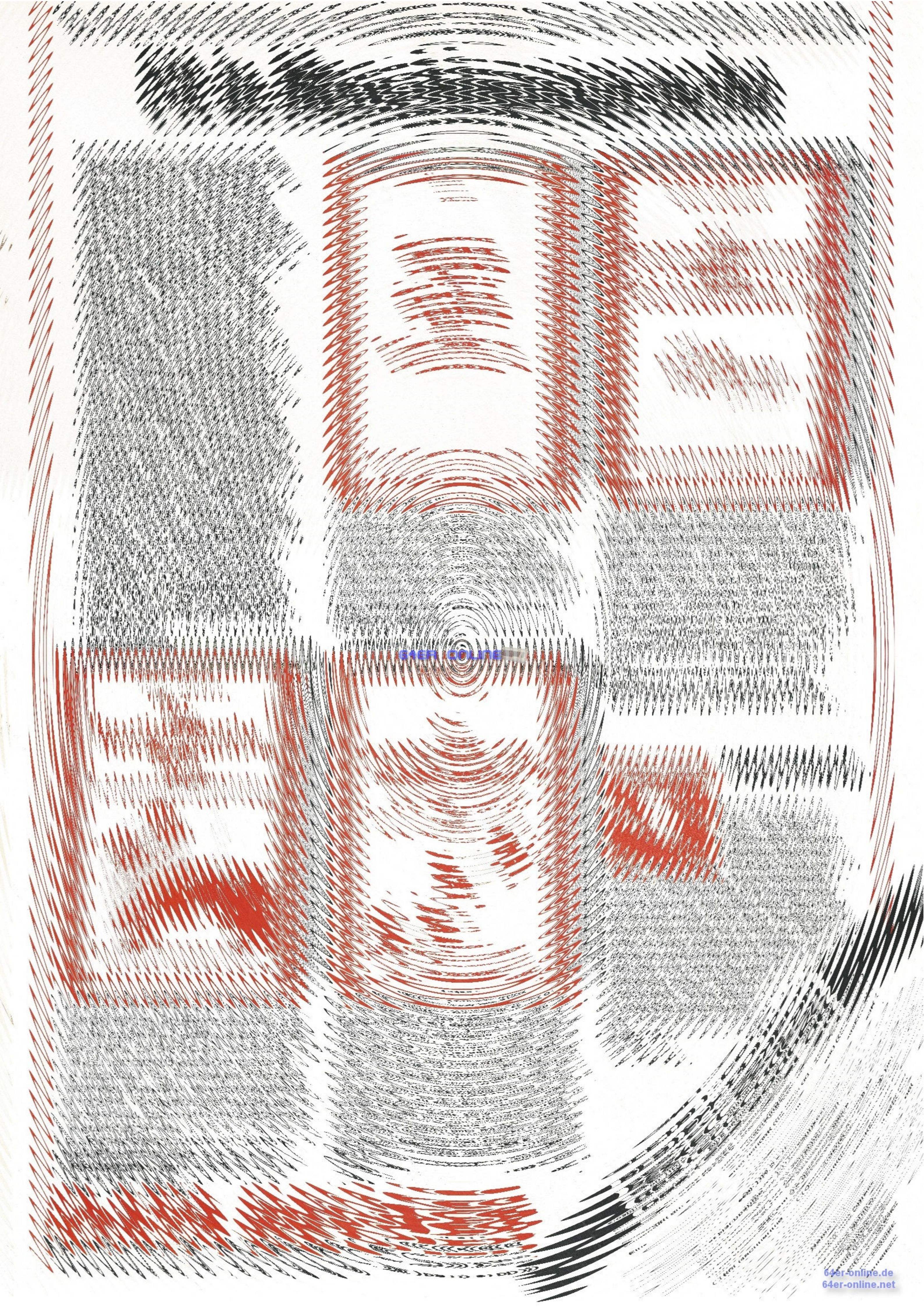
Star unter den Sternen

Vergleicht man Leistungsfähigkeit und die komplette Ausstattung von Star Texter mit dem Preis von 64 Mark, so dürfte es derzeit wohl kaum einen Konkurrenten geben. Zwar ist das zeilenweise Verschieben und Kopieren sicher noch zu verbessern und auch beim Grafikdruck ist noch nicht das Optimum erreicht. Dafür bietet Star Texter aber ein umfangreiches, absturzsicheres und vor allem leicht zu bedienendes Leistungsangebot, das hauptsächlich bei den Sonderfunktionen glänzt. Frei definierbare Zeichensätze, Grafikdruck und Formatmodus sind schon fast Argument genug, um dieses Programm als gut zu bezeichnen. Der 80-Zeichenmodus, der ohne zusätzliches Nachladen bereitsteht, dürfte für ein Programm in dieser Preisklasse wohl einmalig sein. Mit seinem anwenderfreundlichen Preis (64 Mark) und einem fast durchweg gelungenen Konzept dringt Star Texter eindeutig in die Spitzenklasse aller Textverarbeitungsprogramme für den C 64 vor.

(Arnd Wängler/gk)

- Editieren (Zeichen- oder blockweise)
- Kopieren von Textteilen
- Verschieben von Textteilen
- Tabulatoren
- Einfügen, Anhängen von Text von Diskette
- Löschfunktion
- Diskettenbefehle
- Druckereinstellungen
- Finden von Textteilen
- Blättern im Text (schnell und langsam
- Suchen und ersetzen
- Schnittstelle zur Datenverwaltung
- Textformatierung, Trennhilfe
- Blocksatz, Zentrieren
- deutsche Umlaute
- 80-Zeichen-Darstellung
- Insert-Modus
- Rechnen im Text
- Rechnen und programmieren
- Blöcke rollen (horizontal)
- Druckervoreinstellungen
- Grafikdruck (ESC *)
- mehrere Zeichensätze
- Entwurfsblatt für Zeichensätze
- Centronics-Schnittstelle eingebaut
- Cursor-Indikator
- Abschaltautomatik
- Schriftart (ESC !)
- Farbeinstellungen
- verschiedene Tastaturen
- Geräteadresse frei wählbar
- Sekundäradresse frei wählbar
- Parameter abspeichern
- komplette Formatierung
- Reformatierung
- nachträgliche Markierung der Absätze

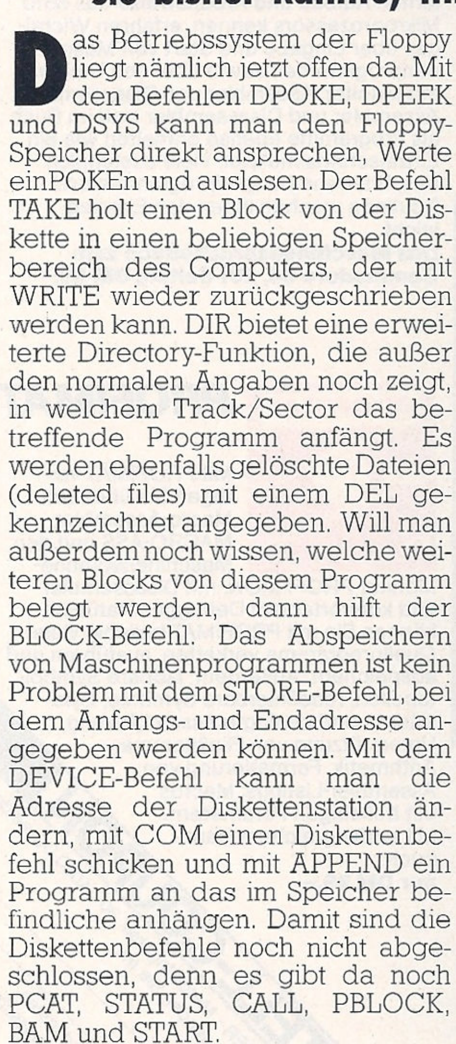
Tabelle 1. Umfassend und vollständig — die Funktionen von Star Texter



Das Intellectool, ein neuer Weg



Einen neuen Weg auf dem von Basic-Erweiterungen überschwemmten Markt beschreitet das Intellectool von GES-Computer. Es ist nicht die Basic-Erweiterung, wie man sie bisher kannte, mit den Grafik, Sound- und Diskbefehlen, es ist mehr etwas für den Freak.



Das Betriebssystem der Floppy liegt nämlich jetzt offen da. Mit den Befehlen DPOKE, DPEEK und DSYS kann man den Floppy-Speicher direkt ansprechen, Werte einPOKEn und auslesen. Der Befehl TAKE holt einen Block von der Diskette in einen beliebigen Speicherbereich des Computers, der mit WRITE wieder zurückgeschrieben werden kann. DIR bietet eine erweiterte Directory-Funktion, die außer den normalen Angaben noch zeigt, in welchem Track/Sector das betreffende Programm anfängt. Es werden ebenfalls gelöschte Dateien (deleted files) mit einem DEL gekennzeichnet angegeben. Will man außerdem noch wissen, welche weiteren Blocks von diesem Programm belegt werden, dann hilft der BLOCK-Befehl. Das Abspeichern von Maschinenprogrammen ist kein Problem mit dem STORE-Befehl, bei dem Anfangs- und Endadresse angegeben werden können. Mit dem DEVICE-Befehl kann man die Adresse der Diskettenstation ändern, mit COM einen Diskettenbefehl schicken und mit APPEND ein Programm an das im Speicher befindliche anhängen. Damit sind die Diskettenbefehle noch nicht abgeschlossen, denn es gibt da noch PCAT, STATUS, CALL, PBLOCK, BAM und START.

Es ist erfreulich, daß das Intellectool über eine eingebaute Centronics-Schnittstelle verfügt, die über den User-Port realisiert ist. Das Anschlußkabel dazu wird mitgeliefert, was den Preis von 228 Mark (ohne Kabel 148 Mark) schon etwas freundlicher macht. Mit diesem besagten Kabel kann man nicht nur die Epson-Drucker hervorragend bedienen, sondern auch andere Drucker.

Da solch hervorragende Hardware angeschlossen werden kann, liegt der Schluß nahe, daß endlich mal grafikmäßig etwas zu machen ist, was über die Bildschirmfähigkeiten des C 64 hinausgeht. Und siehe da, man wird auch in dieser Beziehung vom Intellectool nicht im Stich gelassen. Hat man erst mal durchschaut, welche Sekundäradresse günstigerweise das erwünschte Druckergebnis liefert, dann kann man sich auf gigantische Grafiken stürzen. Mit GFIX legt man (endlich ist's möglich) die horizontale und die vertikale Auflösung fest.

Praktischerweise werden dabei vier Variablen mit besonders seltenen Namen belegt, so daß man vermutlich in keinen Variablenkonflikt kommt. Diese Variablen kann man allerdings sehr gut gebrauchen, zum Beispiel beim TRANSFER-Befehl, mit dem explizit der Grafikspeicher gelöscht wird oder mit einem Muster ganz, oder auch nur zum Teil erfüllt werden kann. Mit POINT setzt man einen Punkt. Linien macht man mit VLINE oder LINE. Wer bei der Verwendung dieser Befehle jetzt allerdings stundenlang wartet, bis sich etwas auf dem Bildschirm tut, der hätte wohl das Handbuch sorgsamer studieren sollen, denn die Grafik wird im Speicher erstellt. Es kann eine virtuelle Grafik mit frei wählbarer Auflösung aufgebaut werden, die bis zu 24 KByte (!) belegen kann. Mit COPY wird sie zu Papier gebracht. Mit COPY dagegen erhält man eine Hardcopy des gegenwärtigen Bildschirminhaltes. Will man dazu auch noch einen Kommentar setzen, dann gibt es da den Befehl für die Leute, die sich nie einer Meinung enthalten, nämlich TYPE. Damit landet man im Schreibmaschinenmodus und schon sagt der Drucker, wozu er per Tastendruck genötigt wird. Wer sich über endlose CHR$-Sequenzen ärgert, den bewahrt der CTRL-Befehl vor dem endgültigen Herzinfarkt. Wer bis jetzt noch nicht verstanden hat, warum die Beispiele im Epson-Handbuch dem C 64 nur einen müden, mitleidigen Syntax-Error abgerungen haben, dem hilft jetzt der LPRINT-Befehl, und wer schon immer schlecht lesen konnte, was der Drucker da so alles ausgibt, der kann jetzt mit DOUBLE die Zeichen in doppelter Größe haben und mit LARGE viermal so groß. Man sieht also, daß für jeden etwas dabei ist. Bei all diesen Möglichkeiten verleitet das Schriftbild, das sich ergibt, wenn man 240 Zeichen/Zeilen anwählt, doch etwas zum Schmunzeln, denn es erinnert etwas an einen Heizkörper und man beginnt seinen Augen zu mißtrauen, wie man das von den softwaremäßigen 80-Zei-

C — Die professionelle Programmiersprache für den C 64

Eine Sprache, deren Programme fast mit der Geschwindigkeit von Assembler laufen, aber mit dem Komfort von Pascal erstellt werden, darf sich zu Recht »professionell« nennen. C für den C 64 ist eine solche Sprache.

Ganz schlicht und einfach »C« heißt die Programmiersprache, die ursprünglich an den Bell Laboratories für das 16-Bit-Betriebssystem Unix entwickelt wurde. C ist eine Weiterentwicklung einer Spezialsprache mit Namen »B« von Digital Equipment, die wiederum auf eine Sprache namens (haben Sie's erraten ?) »A« zurückgeht. B wurde um 1971 geschrieben, um das Betriebssystem Unix — zu der Zeit noch in Assembler programmiert — auf anderen Computersystemen lauffähig zu machen.

1972 erweiterte der bei Bell Laboratories angestellte Programmierer Dennis Ritchie die Sprache B und nannte sie kurzerhand C. Die Vorteile der Programmierung in C gegenüber Assembler waren derart überzeugend, daß 1973 Unix auf C umgeschrieben wurde. Unix war somit das erste Betriebssystem, das in einer höheren Sprache programmiert wurde. Die Tatsache allein, daß es möglich ist, Betriebssysteme in C zu schreiben, wirft ein deutliches Licht auf die Fähigkeiten dieser Programmiersprache.

C wurde von da ab vornehmlich auf 16- oder 32-Bit-Systemen für die verschiedensten Aufgaben verwendet. Für 8-Bit-Prozessoren war C bisher nur für den Z80 (unter CP/M) erhältlich.

Vor kurzem hat nun Data Becker eine C-Version speziell für den C 64 auf den Markt gebracht, den »C-Compiler-64«. Diesen ersten C-Compiler für den 6502/6510-Prozessor haben wir ausführlich getestet. Als erstes stellte sich uns natürlich die Frage, was C an Besonderheiten zu bieten hat und wie die Sprache aufgebaut ist.

Was kann C?

Wer sich an die Sprache C heranwagt, wird sich sehr schnell daran gewöhnen, seine Programme übersichtlich und strukturiert zu schreiben, denn C fordert geradezu dazu auf. C ist von der Syntax her sehr stark an Pascal angelehnt, bietet aber in der Struktur einige Vorteile.

Wie in Pascal können vor Programmbeginn Konstanten vereinbart werden, die sich während des Programmablaufs nicht ändern. Im Unterschied zu Pascal können diese Konstanten aber nicht nur numerische Werte oder Text beinhalten, sondern stehen wahlweise auch für ganze Anweisungsfolgen. Man nennt diese Konstanten dann auch »Makros«. Während der Compilation (Übersetzung) ersetzt der C-Compiler dann diese Makros durch die zugeordneten Anweisungen und bindet sie ins Programm ein.

Auch an Variablentypen hat C einiges zu bieten:

Neben den üblichen Datentypen wie Zeichen (char), 2-Byte-Integer (int) und Fließkommazahlen (float), hat man noch 1-Byte-Integer (short), 4-Byte-Integer (long), Absolut-Integer (unsigned) und doppelt genaue Fließkommazahlen (double) zur Verfügung. Wem das noch nicht ausreicht, der kann sich wie in Pascal auch eigene Variablentypen definieren.

Registervariable

Doch das ist noch nicht alles. Mit C ist es möglich, den Gültigkeitsbereich bestimmter Variablen festzulegen. Ob eine Variable nur in einer Funktion oder für das gesamte Programm gültig sein soll, ist in C schnell und leicht festgelegt.

Wenn es stark auf Geschwindigkeit ankommt, können Variablen auch als Registervariablen deklariert werden. Bei der Übersetzung versucht der Compiler dann, diese Variablen möglichst in den internen Registern der CPU zu halten. Die Verarbeitung erfolgt dadurch um einiges schneller. Diese Register-Option ist natürlich nicht unbedingt für den 6502-Prozessor mit seinen drei Registern (A,X,Y) gedacht.

Besonderes Augenmerk verdient auch die Tatsache, daß C die eigentlich assemblertypische Arbeit mit Zeigern durch eine eigene Arithmetik besonders unterstützt und dadurch auch recht unkonventionelle Wege der Problemlösung erlaubt.

Ebenfalls von Assembler abgeguckt sind die Inkrement- und Dekrementbefehle und die logischen Operatoren, die C zur Verfügung stellt. Neben logischem UND, ODER und EXKLUSIV-ODER hält C auch die Negation, das Einerkomplement und sogar Schiebeoperatoren für Variable bereit.

Selbstverständlich versteht C noch eine ganze Reihe anderer Anweisungen, die in Tabelle 1 zusammen mit den möglichen Datentypen und Speicherklassierungen aufgelistet sind. Betrachtet man diese Liste genauer, wird man viele Elemente der Sprache Pascal wiederentdecken.

Auf den ersten Blick erstaunlich ist, daß C nur relativ wenig Anweisungen versteht. Es zeigt sich hierbei jedoch die C-Philosophie, nur die notwendigsten Funktionen quasi als Grundbaukasten zur Verfügung zu stellen und alle »höheren« Funktionen aus Software- Bibliotheken zu entnehmen. Man wundert sich, welch komplexe Probleme sich zusammen mit der mitgelieferten Funktions-Bibliothek lösen lassen.

Dies ist vielleicht das Bedeutende an der Sprache C. Sie ist aufgrund ihrer wenigen Anweisungen leicht zu lernen, kann aber wegen ihrer Flexibilität zu einer komplexen Struktur aufgebaut werden.

Es ist durchaus möglich, C ohne vorherige Programmierkenntnisse zu erlernen. Hat man jedoch zuvor schon in Assembler oder Pascal gearbeitet, wird man sich sicherlich schneller in C einarbeiten können.

Bild 1 zeigt ein kleines aber typisches Programm in C, das die Fakultät einer Zahl berechnet.

C-Compiler-64

Seit kurzem kann jeder C 64-Besitzer alle geschilderten Vorzüge von C genießen: Mit dem »C-Compiler-64« von Data Becker. Geliefert wird der Compiler im Data Becker-üblichen rot-weißen Ringbuch, das

eine 274 Seiten starke Anleitung und selbstverständlich die Diskette mit den notwendigen Programmen enthält.

Das umfangreiche Handbuch gliedert sich in zwei Teile, den Übungsteil und den Systemteil.

Der Übungsteil beschäftigt sich mit dem Einarbeiten und dem Anwenden der vier verschiedenen Teilprogramme des »C-Compiler-64« in recht ausführlicher Weise. Außerdem enthält er eine Einführung in die Programmierung der Sprache C, die (wie im Vorwort zu lesen ist) das nötige Grundwissen vermitteln soll.

Es ist sicherlich möglich, die Grundbegriffe von C mit dieser Einführung zu verstehen. Die Erklärung der Sprache C geschieht hier aber in sehr komprimierter Weise, so daß die Anschaffung eines speziellen Lehrbuches für den Anfänger doch empfohlen werden muß.

Zur Auffrischung bereits vorhandener C-Kenntnisse leistet der ins Handbuch integrierte C-Kurs allerdings recht gute Dienste.

Der zweite Teil des Handbuchs mit der Bezeichnung »Systemteil« zeigt noch einmal zusammengefaßt die vollständige Funktions- und Anwendungsweise der einzelnen Teilprogramme, wie sie im Übungsteil teilweise sehr ausführlich erklärt wird. Ebenso wie die anschließende Syntaxübersicht und Kurzerklärung der Befehle von C ist diese Zusammenfassung als eine Art Nachschlagewerk zum Compiler gedacht.

Standardbibliothek auf Diskette

Doch nun zur Programmdiskette: Auf der Diskette befinden sich neben dem C-Lader die Teilprogramme C-Copy, C-Editor, C-Compiler und C-Linker. Zusätzlich findet man noch die für C obligatorische Standardbibliothek und einige Beispielprogramme zum Experimentieren.

Zu Beginn hat man den C-Lader zu laden und zu starten. Man findet sich wieder in einem übersichtlichen Menü, über das man die vier Teilprogramme aufrufen kann. Über diesen Lader können auch übersetzte C-Programme geladen und gestartet werden.

C-Copy

Als besonderes Bonbon wird C-Copy im Vorwort des Handbuches bezeichnet. C-Copy ist ein kleines Hilfsprogramm, das die Arbeit mit der Floppystation erleichtern soll. Es ist auch wichtig zum Erstellen einer Arbeitsdiskette für die eigenen C-Programme. Doch dazu später.

Datentypen	Speicher-klassierung	Anweisungen
char	auto	break
int	static	case
float	extern	continue
double	register	default
long		do
short		else
struct		for
union		goto
unsigned		if
typdef		return
		sizof
		switch
		while

Tabelle 1. Die Standard-Anweisungen von C

erron	erroff	nmion
nmioff	qerror	error
exit	fileopen	fileclose
putc	getc	puts
gets	putf	getf
move	cursor	strlen
strcpy	strcmp	strcat
alloc	free	

Tabelle 2. Funktionen von »stdio1.1« (ohne Funktionsargumente)

Über einzelne Kommandozeichen lassen sich Fehlerkanal und Directory einlesen. Durch weitere Kommandos (Größer/Kleiner-Zeichen) kann man Files aller Art außer »REL« laden und wieder speichern. Außerdem besteht die Möglichkeit, die Gerätenummer der Floppy softwaremäßig einzustellen, um von einem Gerät zum anderen kopieren zu können.

Wie man sieht, ist C-Copy also ein recht hilfreiches Werkzeug zur Behandlung von C-Programmen und Files.

Text-Editor der Spitzenklasse

Kommen wir jetzt zum wichtigsten Teilprogramm für den C-Programmierer, zum Eingabe-Editor »C-Editor«.

Mit ihm werden die C-Programme eingegeben und editiert. »C-Editor« bietet hierfür eine überraschend große Anzahl von Kommandos und Befehlen, um dies möglichst komfortabel und einfach zu gestalten.

Zum Ersten fällt sofort der geänderte Zeichensatz ins Auge, der angenehm und gut zu lesen ist. Dies ist keine Spielerei, wie manche denken mögen, sondern notwendig, da C einige Zeichen verwendet (zum Beispiel die geschweiften Klammern), die der Standardzeichensatz des C 64 nicht besitzt. So haben einige Tasten neue Zeichen bekommen, an deren Belegung man sich aber sehr schnell gewöhnt.

Wie viele Textbearbeitungsprogramme hat auch C-Editor einen Kommandoblock am oberen Bildschirmrand, der die Cursorposition, den Filenamen, wichtige Meldungen und eine Tabulatorzeile anzeigt.

Es stehen zwei Textbereiche zur Verfügung, der normale Filetext und ein sogenannter Extratext. Über Tastendruck kann man zwischen den beiden »Textseiten« hin und her blättern.

Es ist möglich, das Format des Eingabetextes zwischen 40 und 80 Zeichen pro Zeile vorzuwählen. Wird ein Format größer 40 Zeichen gewählt, so wird der Bildschirm bei der Eingabe und beim Editieren auf angenehme Weise Zeichen für Zeichen nach links oder rechts weggescrollt. Die Eingabe von Zeilen, die länger als 40 Zeichen sind, ist deshalb sehr komfortabel und übersichtlich (anders als bei manchen Textbearbeitungsprogrammen, die durch sprunghaftes Scrollen die Eingabe von Texten zu einem Abenteuer werden lassen). Während der Programmeingabe kann das Format leider nicht mehr verändert werden, da der Editor dann den kompletten Filetext löscht (new text-Befehl)!

Jede Zeile kann außerdem verschieden eingefärbt werden, was zur Übersichtlichkeit von C-Programmen viel beiträgt.

Wie viele Textprogramme besitzt auch der C-Editor eine sogenannte letzte Zeile (ähnlich wie bei Vizawrite das »end of page«), welches das Textende anzeigt. Leider zeigt C-Editor diese Zeile im Text fatalerweise nicht an. Das Ende des Textbereiches bemerkt man erst, wenn man keine Zeichen mehr eingeben kann und oben im Editorkopf statt des angezeigten Editornamens »C-Editor 1.0« relativ unscheinbar die Meldung »last line« zu lesen ist. So kann das Textende ziemlich überraschend kommen; dann muß man erst wieder mit dem Zeileneinfügekommando (F7) Platz schaffen, da C-Editor dies nicht automatisch macht! Diese Prozedur kann bei längeren Programmen nervtötend sein. Gott sei Dank besitzen aber alle Tasten (so auch F7) eine Repeatfunktion.

Abgesehen davon bietet C-Editor aber sehr viele Kommandos, die eher an ein Textbearbeitungsprogramm als an einen einfachen Eingabeeditor erinnern.

Neben dem Kopieren, Verschieben und Löschen von Textblöcken gibt es auch, wie schon bemerkt, eine Such- und Ersatzoption und ein Einfärbekommando, mit dem man Blöcke in verschiedene Farben tauchen kann.

Schließlich läßt sich auch der Fehlerkanal der Floppy und das Directory einlesen. Doch Vorsicht! Das Directory wird in den Text mit eingebunden und muß dann erst wieder gelöscht werden.

Es empfiehlt sich, vorher auf die Extraseite umzuschalten, wo es nicht stören kann. Natürlich fehlt auch die Printeroption zum Ausdrucken des Textes und die Lade- und Speichermöglichkeit nicht.

C-Editor ist insgesamt sehr komfortabel und steht manchen Textverarbeitungsprogrammen in der Anzahl der Kommandos und besonders deren Ausführungsgeschwindigkeit nicht im geringsten nach, eher ist das Gegenteil der Fall.

Der Compiler

Nun zum Kernstück des »C-Compiler-64«, dem Compiler selbst.

Wie bei vielen Compilern gibt es auch bei diesem für den Anwender wenig zu tun. Die Arbeit leistet nämlich der Compiler selbst, und zwar relativ schnell, wie es bei der Sprache C üblich ist.

Nach der Eingabe des Namens des zu compilierenden Programms und dem Filenamen des compilierten Programms, heißt es für den Programmierer beobachten und hoffen, daß kein Fehler in seinem Programm steckt.

Zu sehen gibt es allerdings nicht viel. Es wird nur angezeigt, bei welcher Funktion oder welchem Zusatzfile der Compiler gerade ist.

Hat er einen Fehler entdeckt, so gibt er eine komplette Fehlermeldung mit betreffender Zeilennummer (wichtig fürs spätere Editieren) auf dem Bildschirm aus.

Ein Fehlerprotokoll auf dem Drucker ist leider nicht möglich.

Man muß jetzt jedoch nicht wie wild alle auftretenden Fehler mitschreiben, denn das übernimmt der Compiler selbst. Während des Compilierens werden alle auftretenden Fehler in einem Extrafile (»error-c«) mitprotokolliert. Dieses File kann später mit dem Editor in den Extratext geladen werden, um dort bei der Fehlersuche und deren Behebung behilflich zu sein.

Gehört man zu den seltenen Glückspilzen, deren Programme keine Fehler haben, ist die Arbeit allerdings noch nicht zu Ende. Denn mit dem vom Compiler erstellten sogenannten Linkfile kann man noch nichts anfangen. Es muß erst vom sogenannten Linker mit anderen notwendigen Programmteilen verbunden und zu einem lauffähigen Programm zusammengebastelt werden. Diese Arbeit besorgt der »C-Linker«, das vierte Teilprogramm des C-Compiler-64.

Der Linker

Dieser Teil des C-Compilers-64 erzeugt, wie schon erwähnt, ein lauffähiges C-Programm aus dem vom C-Compiler gelieferten Linkfile.

Mit ihm können Sie nun getrennt compilierte, aber zusammengehörige C-Programme aneinanderbinden.

Haben Sie Funktionen aus der Standardbibliothek in Ihrem C-Programm verwendet (was meistens der Fall ist), so muß auch das Linkfile der Standardbibliothek dazugebunden werden.

Der Linker sucht sich dann aus der Bibliothek die verwendeten Funktionen und fügt sie dem eigentlichen C-Programm hinzu. Sie können aber auch selbstgeschriebene oder erweiterte Funktionssammlungen mit dem Linker an Ihr C-Programm binden. Wichtig ist nur, daß sich alle notwendigen Files auf der verwendeten Diskette befinden. Deshalb ist es ratsam, vor der Arbeit mit dem Compiler und dem Linker mit C-Copy eine Arbeitsdiskette anzufertigen, die das Linkfile der Standardbibliothek und eventuell eigene Funktionssammlungen enthält.

Mit dem C-Linker können maximal sieben verschiedene Linkfiles zu einem lauffähigen C-Programm zusammengebunden werden. Dabei steht es dem Benutzer frei, zu wählen, ob das fertige C-Programm später von Basic oder nur vom C-Lader aus geladen und gestartet werden soll.

Nebenbei ist es noch möglich, die Speicherobergrenze festzulegen, um eventuell obere Speicherbereiche vor den C-Programmen zu schützen.

Ist das getan, heißt es warten. Je nach Anzahl und Länge der einzelnen Linkfiles kann es mehr oder weniger lang dauern, bis der C-Linker in zwei Durchgängen (Fachwort: passes) alle notwendigen Funktionen und Files zu einem C-Programm »verschweißt« hat, das endlich lauffähig ist.

Selbstverständlich können auch hierbei Fehler auftreten, wenn zum Beispiel eine benötigte Funktion in keinem der angegeben Linkfiles zu finden ist.

Gott sei Dank ist es möglich, den C-Linker wie auch alle anderen Teilprogramme des C-Compiler-64 durch gleichzeitiges Drücken von RUN/STOP und RESTORE zu unterbrechen. Mit einem kleinen Untermenü kann man dann auf Tastendruck einen Neustart wagen oder zum Lader zurückkehren, von dem aus alle anderen Teile des C-Compiler-64 geladen werden können.

Läuft alles zur Zufriedenheit (was selten auf Anhieb passiert), hat der C-Linker das lauffähige C-Programm auf der Arbeitsdiskette abgelegt. Zur Erinnerung, ob es sich um eine B-Version (nur von Basic aus zu laden und zu starten) oder um eine L-Version (Starten nur vom C-Lader« möglich) handelt, sollte man dies im Programmnamen vermerken. Denn eine L-Version läßt sich von Basic aus nicht starten und umgekehrt eine B-Version nicht über den Lader.

Vorsicht ist bei den B-Versionen geboten. Ist das C-Programm beendet, wird sofort ein Reset ausgelöst, und unser Computer meldet sich mit seiner bekannten Einschaltüberschrift. Ein Neustart ist dann nur mit dem erneuten Laden des C-Programms möglich (nervtötend bei umfangreichen Programmen).

Bei der L-Version wird man auch wenig Glück haben. Nach Beendigung des Programms springt der Computer zwar in den Lader zurück, doch warten muß man trotzdem, wenn man neu starten will. Das C-Programm wird nämlich nochmals geladen! Eine Sicherheitsabfrage sollte daher am Ende eines jeden C-Programms nicht vergessen werden.

Positiv dagegen ist, daß ein laufendes C-Programm (ob L- oder B-Version) wie der Compiler selbst mit RUN/STOP und RESTORE unterbrochen werden kann. Dann kann man ebenfalls per Untermenü entscheiden, ob mit dem Programm fortgefahren, wieder gestartet oder dieses beendet werden soll. Das gleiche geschieht, wenn ein sogenannter Laufzeitfehler auftritt. Wird im Programmablauf zum Beispiel eine Variable durch Null geteilt, stoppt das Programm mit der entsprechenden Fehlermeldung und dem bekannten Untermenü, das sich jedoch auch abschalten läßt.

```
page:   1        fakultaet.c
date: 09/07/85
  1 typedef int file;
  2
  3 extern void printf(),scanf(),getc();
  4
  5 char inkey(fd)
  6 file fd;
  7 {  char c;
  8    while((c=getc(fd))==0);
  9    return c;
 10 }
 11
 12 main()
 13 {
 14      int n,i;
 15      double fact;
 16      char z;
 17      z="";
 18      while (z!="q") {
 19        printf("\nDieses Programm berechnet die Fakultaet von n");
 20        printf("\nn= ");
 21        scanf("%d",&n);
 22        fact=1;
 23        for (i=2; i<=n; i++) fact=fact*i;
 24        printf("\nFakultaet von %3d = %f",n,fact);
 25        printf("\nWeiter mit Taste\n");
 26        z=inkey(0);
 27      }
 28 }
 29 extern void erron(),erroff();
 30 extern printf(),exit();
 31
 32 main()
 33 {
 34      int a;
 35      a=0;
 36      erroff();
 37      a=10/a;
 38      a=qerror();
 39      printf("error nr %d",a);
 40      erron();
 41      exit();
 42 }
```


Ein Beispiel zur Programmierung in C. Das Programm berechnet Fakultäten natürlicher Zahlen. Fehler werden abgefangen.

Die Sprache C besitzt in der Grundausführung allgemein keine Eingabe- und Ausgabekommandos. Diese jedoch äußerst notwendigen Routinen sind in der bereits erwähnten Standardbibliothek enthalten. C-Compiler-64 besitzt deren drei:

»stdio.c« (Standard Input/Output) ist ein im C-Quelltext vorliegendes, also noch nicht compiliertes Programm. Es deklariert alle in der Standardbibliothek vorkommenden Funktionen als extern, das heißt als nicht im Quelltext selbst befindlich. Der Compiler muß dies wissen, um ordnungsgemäß übersetzen zu können. Weiterhin enthält dieses File noch wichtige Konstanten.

Die Standardbibliothek

Man kann diese einzelnen Deklarationen auch selbst vornehmen. Es ist jedoch einfacher, dies durch einen C-Befehl («#INCLUDE "stdio.c"«) am Anfang seiner Programme zu tun. Der Compiler bindet dann »stdio.c« an dieser Stelle in den Quelltext ein. Das zweite File »stdio1.l« ist im dritten File »stdio2.l« nochmals enthalten. Man benötigt also eigentlich nur »stdio2.l«. Diese bereits compilierte Bibliothek enthält alle wichtigen Ein- und Ausgabefunktionen, die zur Bildschirm-, Tastatur- und besonders zur weiteren Peripheriebedienung unbedingt notwendig sind. Zusätzlich sind Funktionen zur Stringbehandlung und zur Laufzeitfehlerbehandlung vorhanden.

Es ist zum Beispiel möglich, mit bestimmten Funktionen die automatische Unterbrechung bei Laufzeitfehlern abzuschalten und den aufgetretenen Fehler aus einem Fehlerregister zu lesen, um eigene Fehlerroutinen zu schreiben. Ebenso läßt sich der NMI (Nicht-maskierbarer Interrupt) an- und abschalten und schließlich selbst Laufzeitfehler auslösen, was recht unkonventionelle Problemlösungen zuläßt. Ebenso fehlen auch die für die Sprache C wichtigen Funktionen zur formatierten Ein- und Ausgabe nicht, die verschiedene Formate und dem Standard entsprechend verschiedene Zahlensysteme verarbeiten können.

Insgesamt bietet die Standardbibliothek des C-Compiler-64 von einen recht zufriedenstellenden Funktionsvorrat, der die für C notwendigsten Routinen enthält. Tabelle 2 zeigt den vollständigen Funktionsumfang von »stdio1.l«.

Wer will, kann seine eigene Funktionssammlung aufbauen oder »stdio2.l« durch Zusatzfiles erweitern und so mit der Zeit seine Bibliothek mit Funktionen ausstatten, die sonst nur C-Compiler auf größeren Rechnern besitzen.

C-Compiler-64 beherrscht bis auf Bitfelder den kompletten Sprachumfang von C. Während der Testdauer konnten auch keinerlei Übersetzungsschwierigkeiten festgestellt werden. Allerdings muß man bei der Verwendung von Bibliotheksfunktionen aufpassen, wenn man Programme von anderen C-Versionen abtippt. Es gibt zwar nur selten Dialektunterschiede bei C, doch kann es vorkommen, daß bei bestimmten Compilern gewisse Funktionen andere Namen als üblich haben oder im C-Compiler-64 nicht enthalten sind. Diese müssen mit Makros umdefiniert oder mit vorhandenen Funktionen zusammengesetzt werden.

Fazit

Insgesamt ist Data Becker mit dem C-Compiler-64 eine durchweg gelungene Implementierung der Sprache C gelungen. Bis auf die besagten Schönheitsfehler im ansonsten vorbildlichen Eingabe-Editor steht dem Anwender damit ein durchaus professionell zu nennendes Entwicklungssystem für C zur Verfügung. Natürlich kann man nicht erwarten, alle auf Großrechnern üblichen C-Bibliotheksprogramme auf der Diskette zu finden, bei einem Preis von immerhin 298 Mark hätte es aber schon ein wenig mehr sein dürfen. Für ungeduldige Anwender sind die relativ langen Ladezeiten des Compilers und des Linkers auf Dauer doch recht störend. Es wäre wünschenswert, wenn man sich bei Data Becker dazu entschließen könnte, die Schnellade-Routinen des Profi-Pascal (Test Ausgabe 8/85) auch in den C-Compiler zu integrieren.

Für den Anwender, der sich ernsthaft mit der Sprache C befassen will, ist dieser erste C-Compiler für den C 64 sicher eine interessante Sache. Ganz ohne 16-Bit-Computer oder CP/M eröffnet sich hier die Welt der C-Programmierung. Man sollte aber nicht vergessen, als Anfänger ein geeignetes Lehrbuch zusätzlich zu kaufen, um die Sprache richtig beherrschen zu lernen.

(M. Thomas/ev)

Info: C-Compiler-64, Data Becker, Düsseldorf, 298 Mark

Eintausend und ein

»Schon wieder ein Forth-Test«, werden jetzt sicher manche Leser sagen. Doch mit Super-Forth-64 testeten wir die

Wir haben schon einige Forth-Implementationen für den C 64 vorgestellt. Aber um es gleich vorweg zu nehmen: Sie werden allesamt von Super-Forth-64 in den tiefsten Schatten gestellt.

Super-Forth kann in der Grundversion mit einem Befehlssatz von weit über 700 Worten aufwarten; üblich sind 200 bis 300. Noch dazu erhält man den Source-Code für ein weiteres Hundert Worte.

Große Kompatibilität

Super-Forth-64 entspricht den drei wichtigsten Forth-Standards: FIG-Forth, Forth-79 und der Forth-Version, die Leo Brodie im Forth-Kultbuch »Starting Forth« verwendet hat. Somit kann man fast absolut sichergehen, daß nicht-computerspezifische Forth-Programme nach dem Abtippen problemlos laufen.

Noch dazu verfügt Super-Forth-64 über eine riesige Menge an systemspezifischen Befehlen, die sämtliche Anwendungsbereiche des C 64 wie Grafik oder Sound abdecken.

Natürlich benötigt diese Befehlsvielfalt auch einiges an Speicherplatz: Gut 22 KByte belegt die gelieferte Version. Diese kann man aber sofort einer Abmagerungskur unterziehen und alle systemspezifischen Worte herausschmeißen. Damit wird der Speicherbedarf auf etwa 11 KByte begrenzt. Für alle systemspezifischen Worte, die »High-Level«-Definitionen haben, die also selber in Forth programmiert wurden, wird der Source-Code auf der Diskette mitgeliefert. Somit kann man sich jederzeit nur die Worte in das System einbinden, die man wirklich braucht. Für einige wenige Worte, die »Low-Level«-Definitionen haben, also in Maschinensprache geschrieben wurden, kann kein Source-Code geliefert werden, da sie nicht im Speicher verschiebbar sind. Diese 46 Worte belegen allerdings nur 1,7 KByte und bieten enorme Möglichkeiten, so daß man wohl nicht auf sie verzichten möchte.

Editor und Assembler

Mit Super-Forth-64 werden der Standard-Forth-6502-Assembler und der Forth-Zeileneditor mitgeliefert. Beide Programme haben sich in den letzten Jahren bewährt und sind praktisch bei jeder Forth-Version vorhanden. Also ist auch hier Kompatibilität geboten. Zusätzlich müssen Umsteiger von kleineren Forth-Systemen sich nicht mit neuen, komplizierten Editoren und Assemblern herumschlagen.

Wie sich während der gesamten Testzeit zeigte, ist Super-Forth-64 hervorragend zur Programmentwicklung geeignet. Will man beispielsweise ein Spielprogramm in Forth schreiben, wird man einige Routinen immer in Assembler programmieren. Nun müßte man während der Entwicklung immer die zwei KByte des Forth-Asemblers im Forth-Kern mit sich herumschleppen. Das fertige Spiel würde aber den Assembler nie mehr benötigen. Um diese zwei KByte nun auch noch zu nutzen, kann der Assembler während der Entwicklungsphase in einen anderen Speicherbereich ausgelagert werden und ist im Endprodukt nicht mehr vorhanden.

Peripherie-Bedienung

Ungeschlagen ist Super-Forth-64 auf dem Gebiet der Peripherie.

Super-Forth-64 arbeitet wahlweise mit Kassette oder Diskette als Massenspeicher. Empfehlenswert ist aber auf jeden Fall ein Diskettenlaufwerk. Bei Super-Forth-64 wurde das Konzept des »virtuellen Speichers« konsequent verwirklicht.

So arbeitet der virtuelle Speicher: Zum Schreiben des Source-Codes stehen Ihnen 170 Screens zur Verfügung. Jeder Screen umfaßt 16 Zeilen zu je 64 Zeichen, also genau 1024 Zeichen. Der Speicher zum Editieren beträgt also 170 KByte. Nun passen die aber nicht in die 64 KByte RAM des C 64. Deswegen sind alle Screens auf die Diskette ausgelagert worden. Beim Editieren eines Programms wird nun immer der aktuelle Screen gelesen oder geschrieben. Der Programmierer bemerkt das alles aber gar nicht, denn das Forth-System verhält sich nach außen hin so, als ob alle Screens im RAM vorhanden wären. Man braucht sich beispielsweise nie Sorgen um das Abspeichern von editierten Screens zu machen, weil Forth das automatisch durchführt. Alles in allem ist die Diskette unter Forth nichts weiter als ein virtuelles (scheinbares) RAM.

Der virtuelle Speicher benötigt 680 Blöcke, also eine komplette Diskettenseite. Disketten, die als virtueller Speicher dienen, können niemals zur Speicherung von normalen Files verwendet werden und umgekehrt. Geradezu fantastisch ist noch die Möglichkeit, mehrere Diskettenlaufwerke (bis zu vier) als einen einzigen virtuellen Speicher zu verwenden. Verständlicherweise funktioniert der virtuelle Speicher nicht mit Kassette. Dort werden Screens als normale Files gespeichert, was weitaus unkomfortabler ist, hat man doch auch nur die Screens zur Verfügung, die im tatsächlichen RAM stehen.

Auch die Druckerbedienung ist problemlos. Die entsprechenden Befehle sind stark an die Basic-Kommandos angelehnt, so daß hier kaum Umgewöhnung notwendig ist. Als Source-Code werden sogar Befehlsworte für die RS232-Schnittstelle und den Commodore-Plotter 1520 mitgeliefert. Zum Thema Peripherie gehören eigentlich auch Bildschirm und Tastatur. Auch hier ist volle Unterstützung des Programmierers durch eine Vielzahl von Befehlen gewährleistet. Die Schwächen so manches Basic-Befehls sind dabei noch ausgebügelt worden. Beim C 64 sind schließlich auch noch Joystick, Paddle und Lightpen Peripheriegeräte, die ebenfalls mit Forth-Befehlsworten bedacht wurden.

Grafik und Musik

Man muß es fast schon als selbstverständlich ansehen, daß bei einem solch enormen Sprachumfang die beiden großen Pluspunkte des C 64 beachtet werden: Die Grafik- und Sound-Möglichkeiten.

Super-Forth-64 bietet sowohl in der Grafik- wie in der Sound-Programmierung einen ausgereifteren Befehlssatz als manche erhältliche Basic-Erweiterung.

Fangen wir mit der Grafik an. Zuerst einmal gibt es recht primitive Worte, die die Speicher-Adressen des VIC manipulieren. Mit diesen Worten kann man komfortabel die Sachen erledigen, die im normalen Basic immer einen POKE brauchen: Grafik einschalten, Speicher-

Befehl: Super-Forth-64

wohl beste Forth-Version und gleichzeitig eine der leistungsfähigsten Programmiersprachen für den C 64 überhaupt.

adressen festlegen, Farben einstellen und so weiter. Des weiteren gibt es dann Worte, mit denen Linien, Kreise, Ellipsen und Ellipsenausschnitte gezeichnet werden können. Als Krönung des ganzen sind dann noch die Befehlsworte der aus Logo bekannten Turtle-Grafik definiert.

Nicht unterschlagen werden soll hier, daß es natürlich auch diverse Befehle zur Sprite-Modulation bis hin zum einfachen Sprite-Editor gibt. Sehr interessant ist dabei die für eine Programmiersprache ungewöhnliche Möglichkeit, das Aussehen eines Sprites als Befehlswort zu definieren, was die weitere Programmierung sehr erleichtert.

Die Worte für die Sound-Programmierung stehen denen der Grafik in nichts nach. Unter Super-Forth-64 hat man wirklich über den kompletten SID-Chip Kontrolle. Besonders interessant ist der eingebaute Music-Editor, mit dem Musikstücke besser und einfacher programmiert werden können als mit jeder anderen Programmiersprache.

Interrupts

So manches Programmierproblem kann auf dem C 64 recht einfach gelöst werden, bedient man sich der zahlreichen Interrupt-Techniken. Der kleine Nachteil dabei war immer, daß man Interrupts in Assembler programmieren mußte. Super-Forth-64 bietet nun Worte an, mit denen Interrupt-Programmierung in Forth zum Kinderspiel wird, sei es nun ein Rasterinterrupt, um Text und Grafik gleichzeitig darzustellen, die interruptgesteuerte Meßwert-Erfassung oder das Spielen von Musik im Hintergrund.

Interessant sind auch die zwei mitgelieferten Software-Module, die jeweils eine Diskettenseite Source-Screens beinhalten.

Bei dem einen handelt es sich um ein komplettes Fließkommapaket. Normalerweise wird in Forth nur mit Integer-Zahlen gerechnet. Super-Forth-64 enthält im Sprachkern eine ansatzweise Fließkomma-Arithmetik, die die vier Grundrechnungsarten beherrscht. Diese wird durch das Fließkommapaket ergänzt. Es stehen dann mehr mathematische Funktionen zur Verfügung als es das Commodore-Basic erlaubt, ganz abgesehen davon, daß die Forth-Routinen genauer sind. Zusätzlich können Formeln in algebraischer Schreibweise eingegeben werden. Üblich in Forth ist ja sonst nur die umgekehrte polnische Notation.

Das Fließkommepaket enthält außerdem noch Routinen zur schnellen Matrizenrechnung.

Das zweite Zusatzmodul wird als »Artifical Intelligence Modul« bezeichnet. Mit ihm lassen sich prinzipiell sogenannte Expertensysteme erstellen. Die Möglichkeiten sind hier allerdings recht beschränkt, so gibt es keine Optionen für selbstlernende Systeme. Alles, was diese Systeme können sollen, muß vorher programmiert werden. Insgesamt ist also der Begriff »Künstliche Intelligenz« für dieses Paket etwas hochgegriffen. Trotzdem wird man es im einen oder anderen Anwendungsfall gerne verwenden.

Programmierhilfen

Daß Super-Forth-64 ideal zur Programmentwicklung ist, ist hier schon mehrmals gesagt worden. Dies stellen auch die eingebauten Programmierhilfen unter Beweis. Da wäre zuerst einmal ein rekursiv arbeitender Decompiler. Mit ihm können schon definierte Forth-Worte analysiert werden. Rekursiv ist der Decompiler aus folgendem Grund: Nehmen wir an, Sie untersuchen das Wort A. Nun taucht in dessen Definition irgendwann mal das Wort B auf, das sie ebenfalls nicht kennen. Ein Tastendruck, dann wird B decompiliert, und am Ende der B-Definition wird wieder bei A hinter dem Auftauchen von B weiter decompiliert. Diese Verschachtelung können Sie ruhig noch weiter treiben: Bis zur zwanzigsten Ebene herunter zu gelangen ist gar kein Problem, allerdings befinden Sie sich dann meistens schon auf der Primitiv-Ebene, in der Worte in Assembler definiert wurden.

Da wir gerade bei der Rekursion sind: Auch diese wird von Super-Forth-64 voll unterstützt. Forth-Worte können also ohne weiteres sich selbst aufrufen, solange der Prozessorstack nicht überläuft.

Die zweite wichtige Programmierhilfe in Super-Forth-64 ist der Trace-Modus. Vermuten Sie in einem Wort einen Fehler, compilieren Sie es als Trace-Definition. Dann gibt es bei jeder Ausführung auf dem Bildschirm aus, was es gerade macht.

Es sind noch andere Programmierhilfen verfügbar, so zum Beispiel Stack-Kontrollen.

Besonders wichtig in diesem Zusammenhang ist das Wort APPLICATIONS. Hiermit lassen sich »Load-And-Run«-Versionen von Forth-Programmen erstellen. Laden Sie von Basic aus ein mit APPLICATIONS erstelltes Programm und starten es mit RUN, so wird das zuletzt definierte Wort direkt ausgeführt. Das kann zum Beispiel ein in Forth geschriebenes Spiel sein. Die Forth-Herkunft läßt sich diesem Programm nicht ansehen. Da sich aus ihm heraus Super-Forth-64 nicht rekonstruieren läßt, dürfen diese Programme beliebig weitergegeben werden. Dies gilt natürlich nicht für Super-Forth-64 und die mitgelieferten Source-Screens.

Super-Forth-64 bietet noch einiges mehr, so zum Beispiel Zugriff auf das RAM unter dem ROM, Stringverarbeitung oder Diskettenbefehle, doch uns fehlt leider der Platz, die vielen Vorzüge im einzelnen zu besprechen.

Dokumentation

Über die mitgelieferte Dokumentation braucht man nicht viele Worte zu verlieren: Sie ist sehr umfangreich, genau und, wie leider üblich, in Englisch. Wer aber diese Sprache beherrscht, der wird mit einem wirklich vorzüglichen Handbuch belohnt, das nicht nur die Bedienung, sondern auch das Programm selbst genauestens beschreibt.

Fazit

Eines kann man zum Schluß dieses Testberichts ganz klar sagen: So ein Forth gab es für den C 64 noch nie. Gerade diejenigen, die professionelle Programme in Forth schreiben wollen, werden mit Super-Forth-64 bestens bedient. Die dürften sich auch nicht über den Preis von 380 Mark ärgern. Denn hier bekommt man für sein Geld wirklich einiges an Qualität und Quantität geboten. (Boris Schneider/ev)

Super-Forth-64, 378 Mark, Parsec Research, Drawer 1766-P, Fremont, CA 94538.
Vertrieb in Deutschland: Forth-Systeme Angelika Flesch, Schützenstr. 3, 7820 Titisee-Neustadt

Trends

Wir Europäer werden noch einige Wochen auf das in Amerika schon erhältliche »Ballblazer« warten müssen. Die Programmierer von Lucasfilm (bekannt durch die »Star-Wars« und »Indiana Jones«-Filme) und Epyx griffen so tief in ihre Trickkiste, daß die schnellen 3D-Grafik-Routinen nur auf amerikanischen C 64 mit ihrer leicht höheren Taktfrequenz lauffähig sind.

Die Umschreibe-Arbeiten sind allerdings bereits im Gang. Kurz zur Handlung: Zwei Spieler steuern jeweils einen Gleiter über eine Ebene und versuchen, eine schwebende Kugel in eines von zwei Toren zu befördern.

Die Kugel wird mit Energiefeldern eingefangen, bewegt und wieder weggestoßen. Jeder Spieler hat nur den Ausblick aus seinem Gleiter-Cockpit zur Verfügung.

Billig-Software setzt sich mehr und mehr durch.

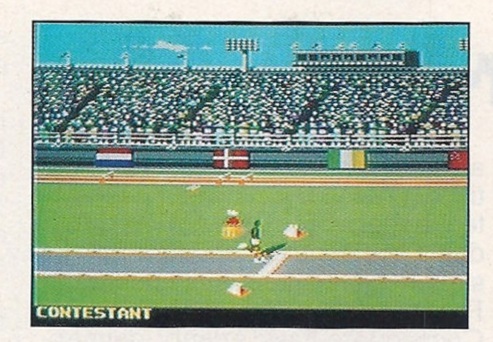
Wir waren recht überrascht, als wir uns die Spiele des Anbieters Mastertronic ansahen. Teilweise hatten diese eine für ihren Preis (12 Mark auf Kassette) sehr gute Qualität. Gefallen haben uns besonders die beiden Action-Adventures »Spooks« und »The Captive« sowie »Kikstart«, ein Motorradrennen für zwei Spieler. Man sollte sich die Mastertronic-Programme ruhig mal beim Händler anschauen, aber auch nicht blind kaufen, es sind schon einige Flops darunter.

Fortsetzungsspiele sind nach wie vor beliebt. Ultimate bietet »Entombed«, den zweiten Teil zu »Staff Of Karnath« an. Taskset hat »Superpipeline II« neu im Vertrieb. First Star arbeitet gerade an einer Fortsetzung zu »Spy vs. Spy«. Und noch dieses Jahr soll »Loderunner III« von Broderbund erhältlich sein.

Die Action-Spiele, auch Schießspiele genannt, scheinen eine Wiedergeburt zu erleben. Gelungene Beispiele sind »Dropzone«, eine sehr gute Defender-Variante, und der bei Redaktionsschluß immer noch nicht lieferbare »Skyfox«. Genaue Tests folgen nächsten Monat.

»Dambusters« hat in England in der Spielehitliste den bisherigen Spitzenreiter abgelöst. Es handelt sich hierbei um ein an sich sehr gut gemachtes Action-Spiel. Allein die Thematik ist uns ein Dorn im Auge. Immerhin wird eine Bombermission auf Deutschland im Zweiten Weltkrieg simuliert.

(M. Kohlen/bs/rg)



Summer Games II

Summer Games II ist nicht etwa ein schlechter Abklatsch des ersten Summer Games, sondern ein sehr gelungener Nachfolger. Die Grafik wurde noch wesentlich besser als die von Summer Games I, und auch die Bewegungen der Akteure wurden realistischer dargestellt.

In Summer Games II wurde besonderer Wert darauf gelegt, daß aus der sportlichen Betätigung kein Joystickkiller wird; vielmehr sind exaktes Timing und eine präzise Joysticksteuerung gefragt.

In Summer Games II können folgende Sportarten gespielt werden: Dreisprung, Rudern, Speerwerfen, Springreiten, Hochsprung, Fechten, Rad- und Kajakfahren.

Da das Spiel erlaubt, auch die Summer Games I-Sportarten einzubinden, hat man, wenn man beide Programme besitzt, 16 verschiedene Disziplinen zur Verfügung. Spielt man mit acht Spielern alle Sportarten durch, so kommt man auf mindestens fünf Stunden Spielspaß.

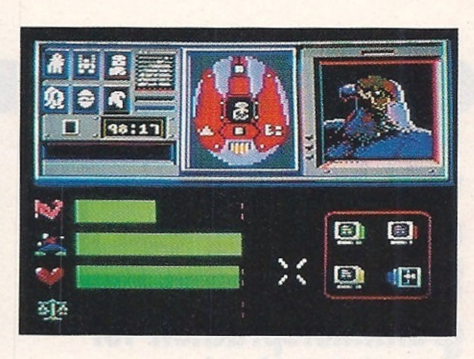
Natürlich kann man Olympiaden auch mit einzelnen Sportarten durchführen. Durch »Compete in some events« ist man in der Lage, sich einen Olympiadurchlauf mit nur den Sportarten zusammenzubauen, zu denen man gerade Lust hat.

Summer Games II ist die ideale Ergänzung zum ersten Summer Games, ist aber auch für sich alleine den Kauf wert.

(M. Kohlen/rg)



Titel	Summer Games II (5 7 9 11 13 15)
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	Sportarten von Summer Games I können übernommen werden
Hersteller	Epyx
Preis	ca. K 59 Mark, D 69 Mark
Bezugsquelle	Rushware



Shadowfire

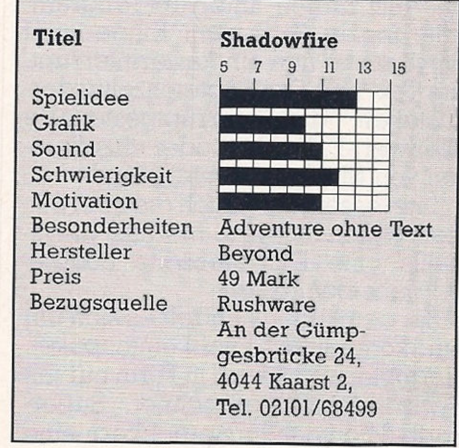
Ein Adventure ganz neuer Art legte Beyond mit »Shadowfire« vor. Die Aufgabe: Innerhalb 100 Minuten ist der entführte Botschafter Kryxix von der Raumstation Zoff V zu befreien. Die Rettungsaktion wird vom Enigma-Team durchgeführt, sechs Personen beziehungsweise Robotern, von denen jeder bestimmte Stärken und Schwächen hat. So kann zum Beispiel nur der Roboter Manto den Bord-Transmitter bedienen und ist somit unbedingt erforderlich, will man nach erfolgreicher Mission zum Enigma-Schiff zurückkehren.

Das Besondere an diesem Adventure ist nicht nur, daß man sechs verschiedene Personen zu steuern hat. Auch die Steuerung selber ist außergewöhnlich: Alle Funktionen werden über Bildsymbole abgewickelt, die auf gut ausgedachte Menüs verteilt sind. Ohne die Anleitung wird man aber kaum die Bedeutung mancher Symbole enträtseln. Spielen kann man wahlweise mit Joystick, Tastatur oder Lightpen.

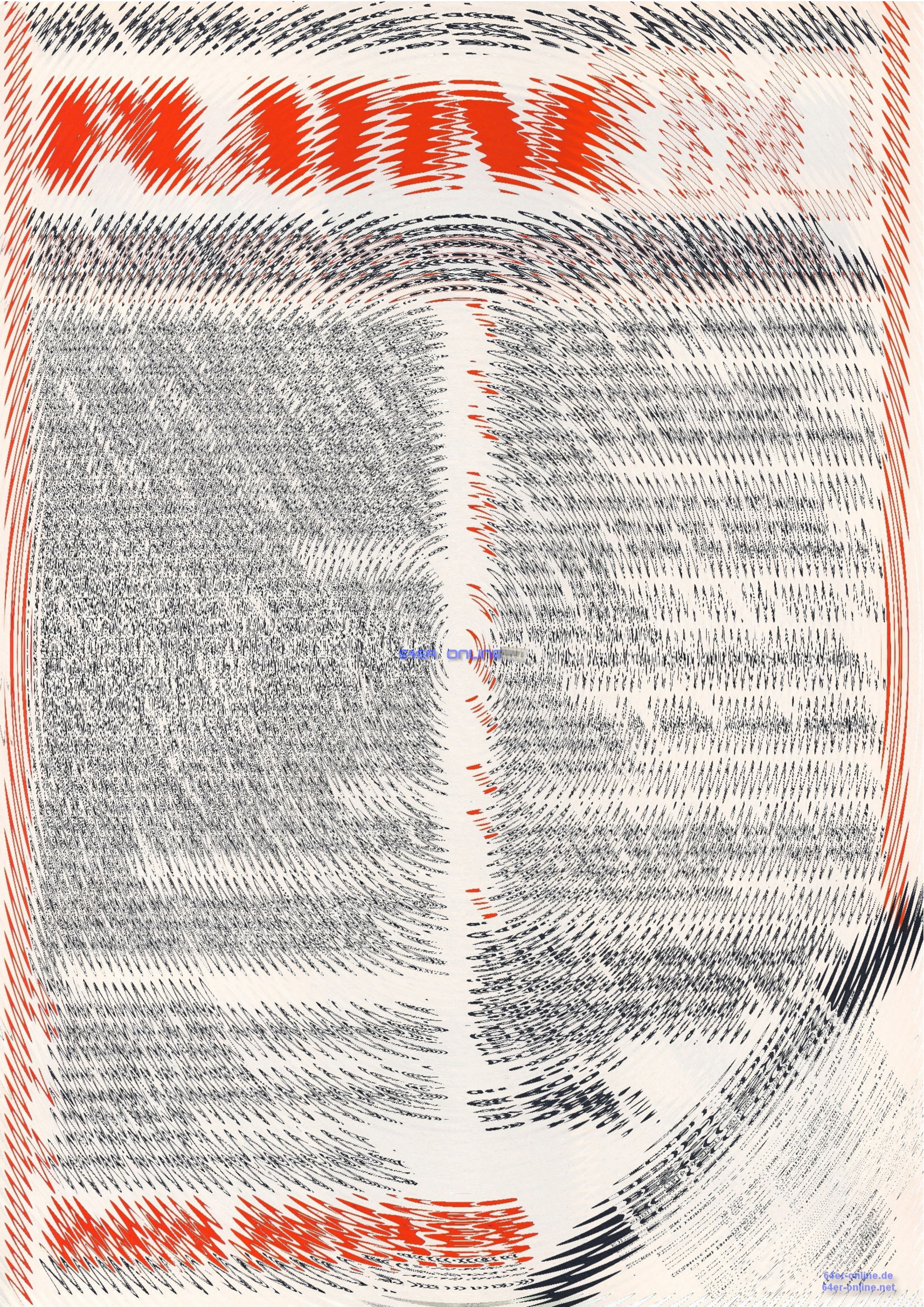
Ebenso außergewöhnlich ist auch die detailreiche Bildschirmgrafik, die allerdings etwas langsam aufgebaut wird, was den Spielfluß ein wenig stört. Gut sind auch die Titel- und die Hintergrundmusik, die sogar in der Lautstärke einstellbar sind.

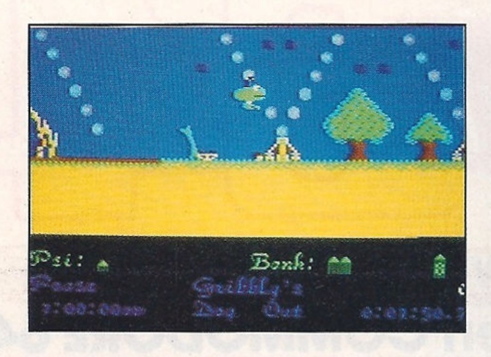
Alles in allem ist Shadowfire ein gelungenes Spiel, das eine Brücke zwischen den Action- und den Abenteuerspielen schlägt.

(bs/rg)



Titel	Shadowfire (5 7 9 11 13 15)
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	Adventure ohne Text
Hersteller	Beyond
Preis	49 Mark
Bezugsquelle	Rushware, An der Gümpgesbrücke 24, 4044 Kaarst 2, Tel. 02101/68499





Gribblys Day Out

Gribblys Day Out ist ein sehr witzig gemachtes Comic-Spiel.

Der einfüßige Gribbly Grobbly muß Gribblets einfangen und an einen sicheren Ort bringen. Zu diesem Zweck kann er hüpfen und fliegen. Beim Fliegen ist er allerdings sehr verletzlich und darf keine Wand berühren, sonst verliert er sofort lebenswichtige Psi-Energie.

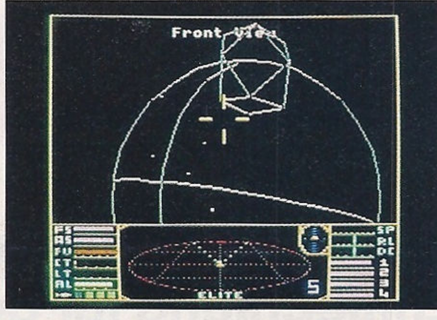
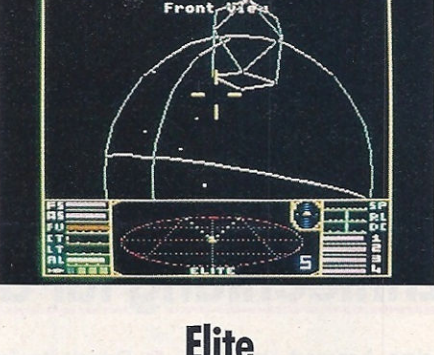
Aber das ist längst nicht alles: Natürlich machen verschiedene Monster die Aufgabe noch schwerer. Sie jagen entweder Gribbly selber oder finden Geschmack an den kleinen Gribblets, deren Vorrat sich so dezimiert. Außerdem sind in der Luft viele Energiesperren in Dreiecksform vorhanden, die Gribbly bei Bedarf aus- und wieder einschalten kann.

Was das Spiel aber besonders spielenswert macht, ist die witzige Animation der Figuren. Gribbly hüpft schöner als ein Känguruh durch die Gegend und lächelt sogar, hat er einen Gribblet gefunden.

Die Gribblets sehen aus wie Pfannkuchen mit Stummelbeinen und die Monster eben wie Monster. Hinzu kommen die sehr großen, unterschiedlichen Screens und die etwas spärliche, aber passende Geräuschkulisse.

Gribblys Day Out werden nicht nur die Kleineren gerne spielen, auch so mancher Erwachsene wird sich wohl verzaubern lassen.

(bs/rg)

Elite

Elite ist ein Spiel, das bisher jeden, einschließlich der Tester, begeisterte. Elite ist eine abwechslungsreiche und sehr komplexe Mischung aus Raumschiff-Simulator, Handels- und nicht zuletzt Action-Spiel. Elite hat eine fantastisch schnelle und gute 3D-Grafik.

Zur Handlung: Sie befahren mit ihrem Raumschiff ein Universum mit über zweitausend (!) verschiedenen Planeten. Um zu Ruhm und Ehre und nicht zuletzt zu Geld zu gelangen, können Sie Handel treiben, aber auch schmuggeln, Raum-Pirat werden oder Kopfgeld-Jäger. Einmalig ist der Detailreichtum, der in diesem Spiel steckt: Da gibt es fast zwanzig verschiedene Raumschiffstypen, vom Piraten- und Polizeischiff bis zur Handels-Raumstation. Gehandelt werden kann mit ebenfalls knapp zwanzig verschiedenen Waren, mit manchen allerdings nur illegal!

Eine enorme Menge an Details bestimmen den Spielverlauf. So können Sie Ihr Schiff ausbauen oder eine von mehreren hochgefährlichen, aber gewinnbringenden Spezialmissionen annehmen. Wer den Titel eines Elite-Kämpfers, der höchste, der verliehen wird, erreichen will und optimal (!) spielt, benötigt mindestens 90 Stunden Spielzeit. Dementsprechend kann der aktuelle Spielstand abgespeichert und später wieder geladen werden.

(bs/rg)

Jump Jet

Zu den beliebtesten Spielen gehören immer noch Flugsimulatoren.

Gerade neu herausgekommen ist die Senkrechtstarter-Simulation »Jump Jet«, für die schon seit Wochen ein riesiger Werbeaufwand betrieben wird.

Das vorliegende Programm ist allerdings etwas enttäuschend. Wir wollen nicht bezweifeln, daß die Simulation des Senkrechtstarters wirklichkeitsgetreu ist, dazu fehlt uns die nötige Flugerfahrung mit einem echten. Was man aber sehen und hören kann, ist sein Geld kaum wert.

Die Bordinstrumente werden aus dem Original-Commodore-Zeichensatz aufgebaut. Der 3D-Ausblick aus dem Cockpit beschränkt sich auf einen immer schnurgeraden Horizont, Wolken und die ab und zu auftauchenden Feindflugzeuge.

Einigermaßen gelungen sind nur die Bilder von der Start- und Lande-Phase, bei der man den Flugzeugträger von oben und der Seite sieht.

Die Titelmelodie ist zwar gut, hat aber extreme Probleme, den Takt zu halten und die eingebaute Sprachausgabe ist die unverständlichste, die je programmiert wurde. Wir haben bis heute nicht herausgefunden, was da gesagt wird.

Interessant dürfte das Programm höchstens für die C 16-Besitzer werden, für die eine eigene Version herauskommen soll.

(bs/rg)



Titel	Gribblys Day Out
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	Auch für Kinder geeignet
Hersteller	Hewson
Preis	29 Mark
Bezugsquelle	Rushware



Titel	Elite
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	über 2000 Planeten
Hersteller	Firebird
Preis	K 79 Mark, D 89 Mark
Bezugsquelle	Rushware



Titel	Jump Jet
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	Senkrechtstarter-Simulation
Hersteller	Anirog
Preis	K 39 Mark, D 49 Mark
Bezugsquelle	Rushware



Fragen Sie doch

Selbst bei sorgfältiger Lektüre von Handbüchern und Programmbeschreibungen bleiben beim Anwender immer wieder Fragen offen. Viel mehr Fragen ergeben sich bei Computer-Interessenten, die noch keine festen Kontakte zu Händlern, Herstellern oder Computerclubs haben. Sie können der Redaktion Ihre Fragen schreiben oder Probleme schildern (am einfachsten auf der Karte »Lesermeinung«). Wir veranlassen, daß sie von einem Fachmann beantwortet werden. Allgemein interessierende Fragen und Antworten werden veröffentlicht, die übrigen brieflich beantwortet.

Auflisten von VERIFY-Unterschieden

Gibt man den Befehl VERIFY ein, so vergleicht der Computer das sich im Speicher befindliche Programm mit dem auf Kassette oder Diskette. Stimmt bei diesem Vergleich auch nur ein Byte nicht überein, so wird eine Fehlermeldung ausgegeben. Meine Frage ist nun, wie man diese Unterschiede, die den Fehler auslösen, auflisten kann (um das Programm anschließend gegebenenfalls zu korrigieren)?

Edith Wehrle

Programme nachladen?

Ich habe ein Basic-Programm geschrieben, das ein anderes nachlädt. Dieses zweite Programm funktioniert für sich alleine tadellos. Wenn ich es jedoch von dem ersten Programm aus nachlade, entsteht ein heilloser Durcheinander im Programm. Wie kann das kommen?

Norbert Burghart

Probleme bei der Hardcopy

Nachdem ich das Programm HI-EDDI aus Ausgabe 1/85 eingegeben hatte, erlebte ich bei der Hardcopy eine böse Überraschung. Der Drucker (BMC BX80) druckte nämlich nur reihenweise irgendwelche Steuerzeichen (wie zum Beispiel Klammeraffen, Grafiksymbole und dergleichen). Was kann ich tun, damit er ausdruckt, was er soll?

Jürgen Bückle

Die Hardcopyroutine von HI-EDDI ist für den Epson FX-80 und Kompatible geschrieben worden. Da der HiRes-Modus beim BMX BX80 über andere Steuersequenzen als beim FX-80 angesprochen wird, muß auch ein großer Teil der Druckerroutine von HI-EDDI neu geschrieben werden.

Vielleicht standen auch einige andere Leser vor diesem Problem und können daher eine passende Lösung für diesen Drucker anbieten?

Programme wiederbeleben nach dem Reset

Wie ist es möglich, Basic-Programme die durch NEW oder einen Reset scheinbar gelöscht wurden, zu restaurieren?

Patrick Künzi

Wir möchten Ihnen zu diesem Problem eine Lösung anbieten, die man sich am besten ausschneiden und an den Computer kleben sollte, denn sie ist sehr kurz, aber wirkungsvoll. Damit ist es möglich, Basic-Programme wieder herzustellen, die entweder durch NEW oder einen Reset gelöscht wurden. Gibt man folgende zwei Zeilen im Direktmodus ein, so kann man das Programm ganz normal weiterbearbeiten:

```
POKE 2050,1
SYS 42291 : POKE46, PEEK (35) :
POKE 45, PEEK (781) + 2 : CLR
```

Adventure-Tips gesucht

1. Wie komme ich bei »Pitfall II« an die Schatulle (das kann auch ein Ring sein)? Ich habe zwar schon einmal alle Schätze, Ratten und Mädchen aufgesammelt, kam jedoch bisher nicht an die Schatulle heran.
2. Wie komme ich bei dem Adventure »The Fabulous Wanda ..« an die richtigen Informationen; wo muß man Geld investieren?

Manfred Wilke

1. Die Lösung zu dem Pitfall-Problem ist ganz einfach: Wenn man das Mädchen »aufgesammelt« hat, muß man fünf Leitern nach unten und dann nach links gehen. Jetzt steht man vor dem Abgrund, über den man hinüberspringen muß. Wenn man dann wieder nach links geht, findet man einen Ring. Daraufhin bewegt man die Figur wieder nach rechts, läßt sich fallen und sammelt weiter.

Sven Steigerwald

2. Betrachtet man das Programmlisting, so erkennt man sehr schnell Sinn und Zweck dieses Spiels, nämlich den Spieler an der Nase herumzuführen! Es ist so aufgebaut, daß man — egal was man macht und wieviel Geld man hat — nie zum Ziel (also einer Nacht mit Wanda) gelangt; man wird immer wieder herausgeschmissen.

Mike Münzberg

C 64-Programme in den C 16 laden?

Wie kann man Programme, die mit dem C 64 auf Kassette gespeichert wurden, in den C 16/116 laden?

Werner Marzilius

Uns ist leider keine Möglichkeit bekannt, C 64- oder VC 20-Software mittels Datasette in den C 16 zu laden. Der einzige Ausweg besteht im Umweg über ein Floppy-Laufwerk.

Computer in die DDR?

Darf man einen kleinen Heimcomputer in die DDR schicken, und wie muß man ihn dort an ein Fernsehgerät (SECAM-Norm) anschließen?

Fritz Ott

Einbinden von SMON

Ich bin sehr erfreut über den Beitrag »Hypra-Ass«, aber leider vermißte ich eine Anleitung zum Einbinden von SMON. Wer weiß Rat?

Boris Schramm

Ich habe mich ebenfalls mit dem Problem befaßt und kam zu folgender Lösung:
1. Man verschiebt SMON, wie in früheren Ausgaben beschrieben, nach $2000 (direkt hinter Hypra-Ass) und lädt danach beide Programme in den Speicher.
2. Damit nun der Quelltext nicht den SMON zerstört, sollte man bei der Intitialisierung von Hypra-Ass eine kleine Änderung vollziehen: Man startet SMON durch »SYS8192« und dissassembliert die Initialisierungsroutine mit D 0B70 0B86 und ändert die folgenden Zeilen ab:
0B70: STA $3000
0B73: LDA #30
0B77: LDA #01
3. Um bei »/b« die richtige Speicherkonfiguration zu erhalten, dissassembliert man: D 1F4A 1F4E und ändert:
1F4A: LDA #30
1F4C: LDX #01
4. Damit man von Hypra-Ass komfortabel in den SMON kommt, wurde dem »X«-Befehl eine neue Bedeutung zugewiesen (es geht ja nicht viel verloren, da der Reset auch leicht durch »SYS 64738« oder durch den Druck auf einen Reset-Taster ausgelöst werden kann.):
O 1E3C 1E3C 1F
O 1E52 1E52 FF
5. Als letztes muß man natürlich alles auch wieder abspeichern, was durch S "HYPRA-MON", 0801 2FFF geschieht.

Peter Schaich

Wollen Sie antworten?

Wir veröffentlichen auf dieser Seite auch Fragen, die sich nicht ohne weiteres anhand eines guten Archivs oder aufgrund der Sachkunde eines Herstellers beziehungsweise Programmierers beantworten lassen. Das ist vor allem der Fall, wenn es um bestimmte Erfahrungen geht oder um die Suche nach speziellen Programmen. Wenn Sie eine Antwort auf eine hier veröffentlichte Frage wissen — oder eine andere, bessere Antwort als die hier gelesene, dann schreiben Sie uns. Antworten publizieren wir in einer der nächsten Ausgaben. Bei Bedarf stellen wir auch den Kontakt zwischen Lesern her.



Probleme mit Netzgrafik

Wie starte ich die Netzgrafik (Ausgabe 4/85) und die zugehörigen Demo-Programme?
Peter Girth

Diese Frage stellen eine ganze Reihe unserer Leser. Deshalb hier eine etwas ausführlichere Antwort, die sich ganz allgemein auf den Umgang mit Maschinenprogrammen bezieht.

Ein Maschinenprogramm, das Sie mit dem MSE eingeben, kann an fast jeder Stelle im Speicher des C 64 liegen. Häufig werden Maschinenprogramme ab Adresse $C000 abgelegt, da dieser Speicherbereich von Basic-Programmen nicht überschrieben werden kann.

Damit das Programm beim Laden wieder an die richtige Stelle im Speicher gelangt, ist Laden mit »LOAD "name",8,1« (»..,1,1« bei der Datasette) erforderlich. Wird die letzte »1« weggelassen, dann wird das Programm automatisch am Anfang des Basic-Bereichs abgelegt, was zwar für ein Basic-Programm richtig ist, für ein Maschinenprogramm in der Regel jedoch die falsche Adresse darstellt.

In derartigen Fällen führt »RUN« meistens zu einem »Syntax Error« und »LIST« liefert nur wirre Zeichen. Apropos RUN: Ein Maschinenprogramm wird sowieso nicht mit »RUN« gestartet, sondern nur mit »SYS«. Die richtige Adresse für den SYS-Befehl müssen Sie der Anleitung zum jeweiligen Maschinenprogramm entnehmen.

Einige Leser beklagten einen »Out of Memory Error«, nachdem Sie das Demo-Programm korrekt mit »LOAD "name",8,1« geladen hatten. Um diesen Fehler zu vermeiden, sollte nach dem Laden eines Maschinenprogramms (aber nicht bei Basic-Programmen !) ein NEW-Befehl gegeben werden. Keine Angst, ein Maschinenprogramm wird durch NEW nicht gelöscht. Dieser Befehl ist bei Maschinenprogrammen aber notwendig, weil der C 64 sich nach jedem Laden die Stelle im Speicher merkt, wo das Programm endet. Bei der Netzgrafik ist das etwa $C780, also weit außerhalb des Basic-Speichers. Das Betriebssystem »vermutet« jetzt fälschlich, daß der gesamte Speicher belegt ist, was zu besagter Fehlermeldung führt.

Ein Wort in eigener Sache

Das Leserforum ist **das** Forum des Gedanken- und Informationsaustausches für Sie, unsere Leser. Hier finden Sie Tips und Tricks für Geräte und Software, hier dürfen Erfahrungen ausgetauscht und Fragen gestellt werden. Damit das Leserforum noch aktueller und interessanter werden kann, bitten wir Sie, die folgenden Punkte zu beachten:

— Wenn Sie Fragen haben, schildern Sie Ihr Problem bitte möglichst umfassend und genau. Allgemeine Fragen wie zum Beispiel »Wie schließe ich eine 20 MByte-Festplatte an den C 64 an«, lassen sich halt beim besten Willen auch nur ebenso allgemein beantworten: »Mit einem Interface«.

— Wenn Sie uns eine Antwort auf eine im Leserforum veröffentlichte Frage schicken, dann sollte Ihr Schreiben bitte unbedingt die Ausgabe des 64'er-Magazins, in der die Frage veröffentlicht worden ist, sowie den Namen des Fragestellers enthalten, dazu möglichst auch noch eine Wiederholung der gestellten Frage in Kurzform. Sie erleichtern uns die Arbeit damit enorm.

— Um den Kontakt zwischen unseren Lesern noch mehr zu verbessern, werden wir in Zukunft auch vollständige Adressen abdrucken. Voraussetzung ist natürlich Ihr Einverständnis. Falls Sie also im Falle einer Veröffentlichung im Leserforum mit der Angabe Ihrer vollständigen Adresse einverstanden sind, dann fügen Sie Ihrem Schreiben bitte eine schriftliche Einverständniserklärung (mit Unterschrift) bei. Auch für eine von Ihnen eventuell gewünschte Weiterleitung Ihres Schreibens an einen Fragesteller aus dem Leserforum benötigen wir Ihr schriftliches Einverständnis.

— In letzter Zeit häufen sich anonyme Zuschriften an das Leserforum, fast ausschließlich mit Phantasienamen gezeichnet, die an die Raubkopierer-Szene erinnern sollen. Dazu nur folgendes: Anonyme Schreiben werden im Leserforum weder veröffentlicht noch sonstwie beachtet. Wenn Sie Wert darauf legen, daß selbst Ihr Name nicht abgedruckt wird, dann teilen Sie uns das bitte im Anschreiben mit — in begründeten Einzelfällen kann dann für den Abdruck der Name von der Redaktion geändert werden. Zuschriften ohne Absender gehen allerdings — sorry — sofort in den Papierkorb. (ev)

Leser fragen — Willi Brechtl antwortet

Hallo liebe Leser, hier bin ich wieder, um Eure Fragen zu beantworten.

Ich werde mich hauptsächlich um Leserbriefe kümmern, die nicht in das sachliche Einerlei des Leserforums passen. Zum Beispiel Fragen, die sich aus dem einen oder anderen Grund nur ganz subjektiv beantworten lassen. Oft genug tauchen auch Probleme auf, die sich nicht mit einem kurzen Antwortsatz abhandeln lassen. Und wenn selbst eine längere Antwort im Rahmen des Leserforums nicht mehr ausreichen würde, dann ist das ganz klar ein Fall für Willi Brechtl.

Also: Wenn Sie als Anfänger Probleme mit Computer, Software oder Handbuch haben, dann wenden Sie sich in Zukunft doch einfach vertrauensvoll direkt an mich.

MPS 802 umsonst

Laut Handbuch hat der »Hochleistungs-Druckkopf« des MPS 802 eine Lebenserwartung von 50 Millionen Zeichen. Nun, meiner war bereits nach zwei Monaten (bei mäßiger Beanspruchung) defekt. Die oberste Nadel verabschiedete sich während des Druckens unauffällig. Hilfesuchend wandte ich mich an ein Fachgeschäft. Dort wurde mir mitgeteilt, daß der gesamte Druckkopf ausgetauscht werden müsse.

Nach mehr als zwei Monaten Lieferzeit kam die Überraschung: Der neue Druckkopf wurde mir mit 378 Mark in Rechnung gestellt. Die Reparatur kostete inklusive Mehrwertsteuer 528,96 Mark.

Nun meine Überlegung: Wenn ich meinen gebrauchten MPS 802 für 530 Mark verkaufen kann, deckt das gerade die Reparaturkosten.

Fazit: Vor der Reparatur muß der Wert des Druckers praktisch 0 Mark gewesen sein. Oder sehe ich das falsch?
Willi Wendt

Lieber Namensvetter!

Das von Ihnen angeschnittene Problem tritt nicht nur im Zusammenhang mit gebraucht gekauften Geräten auf. Auch ein nagelneuer Drucker (und andere Geräte) können schon kurz nach dem Kauf fehlerhaft werden. Für solche Fälle gibt es die Garantie. Nach deutschem Recht muß diese mindestens 6 Monate betragen. Danach ist man auf den Reparaturservice des Händlers, Herstellers oder einer freien Werkstätte angewiesen. Leider funktioniert dieses ganze System nicht so einwandfrei wie man sich das wünscht.

Das Hauptproblem liegt, wie fast immer in der Betriebswirtschaft, in einer Differenz zwischen Angebot und Nachfrage und den damit verbundenen Reparaturkapazitäten. Die Werkstätten der großen Hersteller und Händler haben alle Probleme, mit den sprunghaft angestiegenen Verkaufszahlen des C 64 mitzuhalten. Ihre Kapazitäten waren (und sind teilweise noch) auf wesentlich weniger Geräte ausgelegt.

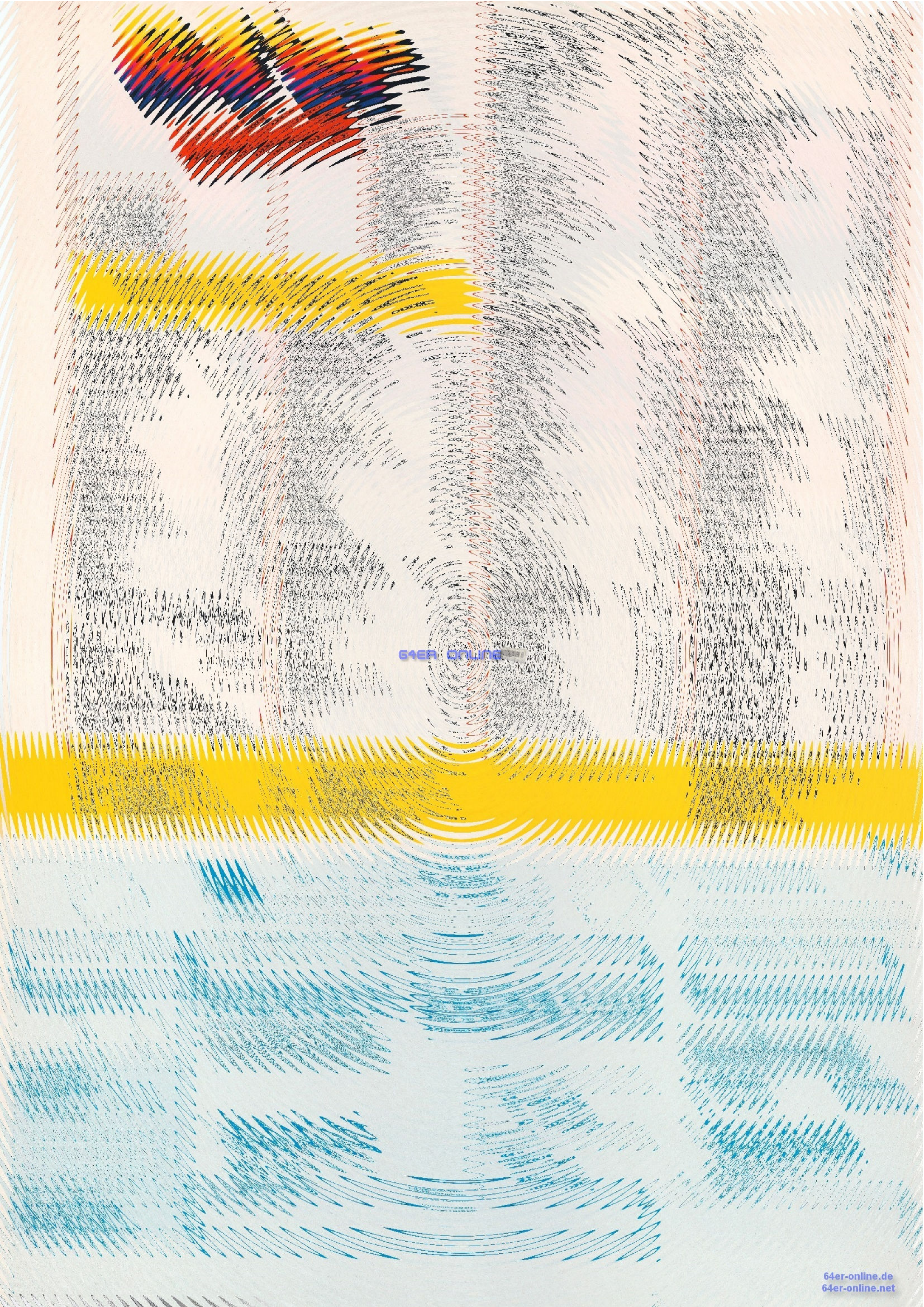
Außerdem gab es mit der Erfindung des Heimcomputers erstmals Computersysteme, die nicht von einem speziellen Service-Personal gewartet wurden, sondern von Menschen wie Sie und ich. Das sind wahrscheinlich die Gründe, warum Sie so lange auf Ihren Drucker gewartet haben.

Die Ursache für den, auch in meinen Augen, etwas »happigen« Preis des Druckkopfes liegt ganz woanders. Drucker waren noch vor einem Jahr wesentlich teurer. Damals wurden Verkaufspreis und Ersatzteilpreise gemeinsam kalkuliert. Mittlerweile hat aber der Markt, mit seinen Gesetzen, zu einer radikalen Preissenkung bei Druckern geführt. Da die Ersatzteilpreise aber nicht den Bedingungen des Marktes unterliegen, bestand auch kein Anlaß, sie zu reduzieren (wer fragt beim Kauf schon nach dem Preis eines Druckkopfes).

Sicherlich könnte man diese Gedanken noch weiterführen (deshalb rufe ich alle, die ähnliche Probleme haben, auf, mir zu schreiben). Aber nun noch etwas zu Ihrem speziellen Problem.

Der Druckkopf des MPS 802 ist ein weit verbreitetes Massenprodukt, der auch im C80X, im DP80 und im BX80 eingebaut ist. Es ist prüfenswert, ob auch diese Hersteller ähnliche Preise für den Druckkopf verlangen. Der Umbau ist in jedem Fall problemlos, er dauert kaum 10 Minuten.
Willi Brechtl







Play by Mail und Play by Modem

oder: Wie man die Einnahmen der Bundespost durch Computerspiele steigert.

Man kann Computerspiele auch spielen, ohne überhaupt einen Computer zu besitzen. Das Zauberwort heißt Play by Mail oder kurz PBM. In den USA hat sich eine große PBM-Industrie entwickelt, die Briten betreiben es meist semiprofessionell, und Deutschland ist noch im Anfangsstadium: Hierzulande spielt sich das alles noch hauptsächlich im Hobbybereich ab.

Die neuere Variante dieser Spielgattung nennt sich ebenfalls PBM. Hier steht diese Abkürzung für »Play by Modem«; den heimischen Verhältnissen angepaßt müßte man eigentlich PBA sagen. A steht natürlich für Akustikkoppler.

Computerspiel per Post

Wer PBM-Spiele eigentlich erfunden hat, weiß keiner so genau. Angefangen hat es wohl mit Briefschach, das die Grundidee lieferte. Allerdings nur mit zwei Spielern gegeneinander. Von Computern war noch nicht die Rede. Im Lauf der Zeit entwickelten sich neue Spielarten, die sehr große Verbreitung fanden. Nach dem Zweiten Weltkrieg wurden Strategiespiele wieder populär, und Anfang der siebziger Jahre begann dann der Boom der Rollenspiele. Die Fans dieser Gattungen trafen sich, da man in Gruppen spielt. Doch einige, die sich dafür interessierten, hatten keine Gleichgesinnten in ihrer Umgebung. Bis jemand etwa Mitte der siebziger Jahre darauf kam, solche Spiele per Post durchzuführen. Die Teilnehmer sendeten ihre Spielzüge an den Spielleiter, der diese dann auswertete und das Ergebnis an die Spieler zurückschickte. Aus diesen Privatinitiativen wurden dann sehr schnell einträgliche Geschäfte, die Zahl der Mitspieler wuchs, und die Veranstalter konnten die Informationsflut nicht mehr manuell verarbeiten. So wurden dann Computer eingesetzt, und das computer-moderierte Briefspiel war geboren.

Vorteile der Briefspiele: Erstens braucht nicht jeder, der mitspielen will, einen Computer. Zweitens ist diese Art von Spielen oftmals viel interessanter als ein Heimcomputerspiel, denn so ein Großcomputer, wie ihn manche PBM-Firmen benutzen, kann schon eine ganze Menge mehr an Informationen verwalten und somit realistischere Spielabläufe simulieren. Und drittens spielt man nicht alleine; in den USA gibt es sogar Spiele, in denen bis zu 1000 Akteure mitwirken. Und den Ansporn, sich mit Leuten, die teilweise aus der ganzen Welt kommen, in ei-

nem Spiel zu messen, kann ein normales Computerspiel nicht bieten.

Die PBM-Spiele können grundsätzlich in drei Gruppen der Computerabhängigkeit eingeordnet werden.

Gruppe 1: Das gesamte Spiel wird durch einen Computer verwaltet und ausgewertet. Die meisten kommerziellen PBM-Anbieter gehören in diese Kategorie; Mammutspiele mit Hunderten von Glücksrittern durch Menschen auswerten zu lassen, ist praktisch unmöglich.

Gruppe 2 arbeitet völlig ohne Computer und ist fast nur auf dem Hobbybereich anzutreffen. Die Auswertung des Spiels ist dann zwar nicht mehr so unpersönlich wie ein Computerausdruck, dafür aber ist so etwas nur mit wenigen Mitspielern machbar und kann weitaus weniger komplex und interessant ausfallen.

Gruppe 3 ist die interessanteste: Die Auswertung erfolgt zwar durch Computer, dieser optimalen Informationsverarbeitung wird jedoch vom Spielleiter ein menschlicher Faktor hinzugefügt und das Ergebnis als individuelles Schreiben dem Teilnehmer zugesandt.

In USA und England gibt es schon PBM-Spiele in großen Mengen. Beispiele sind Silverdawn (das meistgespielte in den Vereinigten Staaten), Duelmasters, Starquest, Moneylender oder Starglobe. In Deutschland ist dieses Hobby noch relativ unterrepräsentiert. Es gibt nur einen einzigen kommerziellen Anbieter (am Rande des Amateurdaseins) und einige Privatanbieter. Beispiele sind Feudalherren (Wirtschafts- und Strategiespiel), Starweb (Science-fiction-Strategie), Diplomacy (historisches Spiel) oder United (eine strategische Fußballsimulation mit allen Raffinessen).

Zwei der Anbieter betreiben die Auswertung ihrer Spiele sogar mit einem C 64. »Nomaden«, programmiert und angeboten von Wolfgang Sommer, ist ein Spiel, in dem es um Besitztümer, Handel, politischen Einfluß und die 13 Steine der Weisen geht. »Ashes of Empire«, von Harald Topf und M. J. Costello, ist ein Science-fiction-Spiel, in dem es gilt, als erster 15 Planeten zu erobern. Besonderer Gag: Man kann das Programm auch kaufen und selbst einen 8-Spieler-PBM-Betrieb veranstalten (Selbstkostenpreis 28 Mark).

Die meisten PBM-Veranstalter in Deutschland bieten ihre Spiele über Insider-Magazine, die sogenannten »Hobbyzines« an. Da wir im Rahmen einer Computerzeitschrift nicht so ausführlich auf solche Randgebiete des Computerspielens eingehen können, haben sich die Autoren dieses Artikels einen besonderen Service einfallen lassen: Wer mehr Hintergrundinformationen, Adressen von Anbietern und Hinweise auf Fachliteratur zum Thema PBM haben will, kann diese kostenlos erhalten bei: Ferdinand Wlodarczyk, Prager Str. 26, 8000 München 45.

Da die Hauptanwendung der computer-moderierten Briefspiele im Bereich Fantasie und Strategie liegt, ist dieses Thema in unserer Information auch kurz angesprochen. Eine neuere Variante dieser Spielgattung ist das Modem-Spiel. Im Gegensatz zum Briefspiel ist hier jedoch der Computer auch für die Mitspieler notwendig.

Modem-Wahnsinn

Die Geschichte »Wargames«, in der der junge Matthew Broderick mit dem Pentagon-Computer 3. Weltkrieg spielt, ist gar nicht so abwegig. Auf die Wirklichkeit bezogen könnte er zwar keinen Krieg auslösen, aber hätte doch immerhin über sein Modem an der Simulation teilgenommen.

Es gibt tatsächlich eine 3. Weltkrieg-Simulation, die mit bis zu acht Spielern am »Livermore Combat Simulation Laboratory« gespielt wird. Das allerdings auch nur von ausgesuchten Mitgliedern des US-Militärs. Wie können wir nun an solchen Spielen teilnehmen?

Um das US-Superding nachzuahmen, bräuchte man immerhin 2000 C 64, die dann nur noch etwa mit der 70fachen Geschwindigkeit laufen müßten. Denn die vielen Details passen nicht in das RAM eines Heimcomputers, und auch die unglaublich hohe Grafikauflösung dieses Mammutprojekts ist kaum auf einem Kleincomputer zu realisieren. Aber: Nachdem in den USA die ersten Netzwerke für Heimcomputer gegründet wurden, ließen derartige Spiele nicht lange auf sich warten. Eines der meistgenutzten Netzwerke der USA, CompuServe, erweckte 1979 mit Decwars das erste öffentlich zugängliche Modemspiel zum Leben. Decwars ist ein Strategiespiel, und obwohl es schon sehr ausgeklügelt und umfangreich war, wurde es noch zu »Megawars« weiterverarbeitet und verbessert und ist noch immer das gefragteste Modemspiel. Das CompuServe-Netz bietet hauptsächlich Strategiespiele. In »Seawars« kann man nicht nur Krieg spielen, sondern einen »Raum« besuchen, in dem man direkt (online) mit den anderen Spielern seine Züge beraten kann. Andere Netzwerke wie »The Source« und »Delphi« haben sich auf traditionelle Brett- und Kartenspiele eingerichtet, also Bridge, Schach und ähnliches.

Die meisten dieser Netzwerke sind hauptsächlich Informationsvermittler und bieten ihre Spiele nur als Nebenbeiservice an. Das Plato-Netzwerk macht es umgekehrt: Der Mailboxbetrieb ist nebensächlich, als Hauptzweck werden 25 verschiedene Spiele angeboten; die meisten davon aus dem Fantasie-Rollenspiel-Bereich.

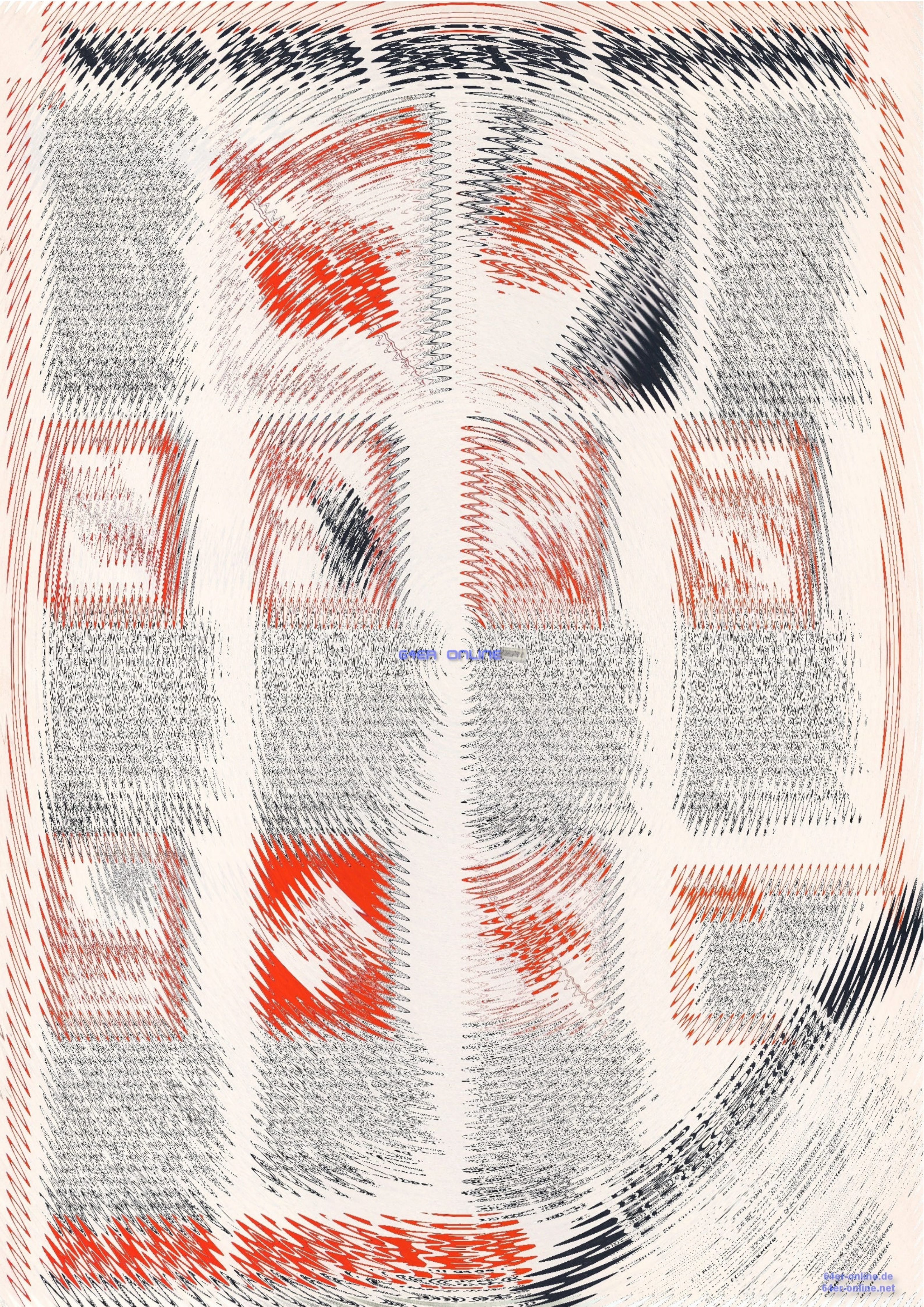
Mittlerweile gibt es sogar schon reine sogenannte »Game services«. Eines davon ist PlayNet, das speziell für C 64-Benutzer gedacht ist. PlayNet überträgt sogar hochauflösende Bilder, in denen (zumeist) Denkspiele verabreicht werden. Also Backgammon auf einem richtigen Backgammon-Feld und nicht als bloße Textübertragung. Außerdem können sich die PlayNet-Besitzer noch während des Spiels unterhalten. Schade, daß es hier bei uns so etwas noch nicht gibt.

Den größten Fortschritt in Sachen Modem-Spiele erreichte in England die Universität von Essex. MUD ist die Abkürzung für »Multi-User-Dungeon« und wurde 1983 auf einem DEC 10 entwickelt. Mit MUD entstand das erste Multi-User-Spiel, denn 38 Leute können gleichzeitig spielen. Und sie spielen nicht alle für sich gegen den Computer, sondern gemeinsam oder gegeneinander. MUD ist ein Rollenspiel: Man läuft in einem riesigen Land herum und erlebt Abenteuer. Was das Spiel so interessant macht, ist der unberechenbare Faktor: Man weiß nie, ob man auf eine vom Computer gesteuerte feindliche Gruppe von Abenteurern stößt oder ob man anderen Modemspielern gegenübersteht.

Im Vergleich zu MUD sind die Adventures, die wir gewohnt sind zu spielen, nur einfache Suds — Single User Dungeons.

MUD kann mit jedem Computer zwischen Mitternacht und sieben Uhr morgens an der Essex-Universität »angezapft« werden (tagsüber wird der Computer für wissenschaftliche Zwecke benötigt). Mittlerweile wurde MUD auch ins englische CompuNet übernommen, einem Netzwerk speziell für Commodore-Benutzer. Dazu allerdings braucht man ein spezielles Compu-

Fortsetzung auf Seite 156



Fortsetzung von Seite 154

Net-Modem; das Modem hat eine Identitätsnummer, die vom Compu-Net-System geprüft wird, so daß man MUD mit einem normalen Modem oder Akustikkoppler nicht spielen kann. Leider. Das Compu-Net-Modem läuft mit 1200/75 Baud und kostet 99.99 englische Pfund. Im Preis inbegriffen ist die Mitgliedschaft für das erste Jahr.

Wer einen längeren England-Aufenthalt plant, sollte unbedingt von CompuNet Gebrauch machen. Er kann dabei nicht nur tage- und nächtelang MUD spielen, sondern hat Zugriff auf das interessanteste Netzwerk, das es für Commodore-Besitzer gibt. Von Deutschland aus kann man leider wegen der Baudrate und der ID-Nummer noch nicht auf CompuNet zugreifen.

Hier in Deutschland gibt es weder spezielle Spielnetzwerke noch Multi-User- oder Online-Spiele. Um die Begriffe kurz zu erläutern: Ein Multi-User-Spiel kann von mehreren Leuten gleichzeitig gespielt werden, wogegen Single-User-Spiele nur hintereinander oder abwechselnd von den Benutzern gespielt werden können. Schach per Akustikkoppler ist beispielsweise ein typisches Single-User-Spiel. Der Unterschied zwischen Online- und Offline-Spielen ist wieder etwas ganz anderes. Bei Offline-Spielen wird der Computer nur als Hilfsmittel benutzt. Der Spieler hackt also seinen Spielzug in eine Mailbox. Hat der Sysop die Züge aller Spieler beieinander, werden sie ausgewertet und durch die Mailbox wieder an die einzelnen Spieler weitergegeben.

Die Entwicklungen in Sachen Play-by-Modem sind in Deutschland noch im Anfangsstadium. Doch es tut sich immerhin etwas. Die Software-Express-Mailbox richtet gerade eine Schachecke ein — das Spiel, das auch schon das PBM-Business zum Leben erweckte. In München bietet die Orbit-Mailbox (sie läuft übrigens auf einem VC 20 mit 64K-Erweiterung) interessierten Leuten die Möglichkeit, das Fantasie-Rollenspiel »Advanced Dungeons & Dragons«, das Fantasie-Fans sicher ein Begriff ist, zu spielen. Bisher sind das alles nur Single-User-Offline-Spiele. Jetzt in Deutschland ein Multi-User-Spiel zu erhoffen, ist noch verfrüht, denn erstens erfordert Multi-User-Fähigkeit größere Systeme wie einen DEC oder eine VAX, und zweitens wird so etwas sicher an den Restriktionen der Bundespost scheitern (Multi-User-Schaltungen müssen angemeldet und genehmigt werden). Größere Systeme gibt es zwar auch hier, doch die Universitäten, in denen sie stehen, würden wahrscheinlich ihre Schranken nicht dem »Modem-Wahnsinn« öffnen. Hoffen wir auf Initiativen vom Chaos Computer Club, der in Deutschland noch immer der Vorreiter in Sachen DFÜ ist.

Online-Spiele wären bei uns allerdings durchaus möglich: Man nehme einen C 64 und ein Adventurespiel (kann sogar in Basic sein, das spielt bei 300 Baud keine Rolle). Die Tastatureingabe und die Bildschirmausgabe lenke man auf die RS232-Schnittstelle um, und schon hat man sein Single-User-Online-Spiel.

Wir wollen hiermit alle Mailbox- und Telefonservice-Anbieter aufrufen, eventuelle Entwicklungen und Angebote auf diesem Sektor an die 64'er-Redaktion zu melden. Sobald sich einiges getan hat, werden wir über die neuesten Entwicklungen berichten und dabei auch auf Btx-Spiele näher eingehen.

(M. Kohlen/rg)

Info: Für die Versorgung mit Unmengen Material zu Computerbriefspielen möchte ich folgenden Personen danken:
Ferdinand Wlodarczyk
Joachim Durchholz
Frank Grabert
Wolfgang Sommer
Dragon (Fantasy-Magazin)

Achtung, Interessierte

Da wir in unserer losen Folge von Spiele-Grundlagen weiterhin über interessante Spiele-Arten, Formen des Computerspiels und zukünftige Entwicklungen berichten wollen, bitten wir alle, die Anregungen und Informationen bieten können, diese an die Redaktion zu senden.

Da wir Anfang nächsten Jahres einen größeren Bericht über Simulationsspiele planen, wollen wir speziell zu diesem Thema gezielte Informationen sammeln. Wer über ernsthafte Simulatoren (wie beispielsweise die der Lufthansa oder des VW-Werks) berichten kann, sollte sie bitte an die 64'er-Redaktion schicken; möglichst mit Fotos und Beschreibung. Wir hoffen, durch Ihre Mitarbeit vollständige Hintergrund-Informationen liefern zu können.

Bitte senden an:
Markt & Technik
Redaktion 64'er
Christian Rogge
Hans-Pinsel-Str. 2
8013 Haar

James B

Jetzt ist es da, das Spiel zum neuen James-Bond-Film, der in diesen Tagen in den deutschen Kinos angelaufen ist. Das heißt eigentlich sind es ja drei Spiele, die zwar aufeinander aufbauen, aber unabhängig voneinander gespielt werden können.

Schlüpfen Sie in die Rolle des Agenten 007 und lösen das Geheimnis um Max Zorin, einem europäischen Elektronik-Magnat. Wie vor jedem James-Bond-Film ist auch vor Beginn des Spieles der bekannte Vorspann zu sehen. Dann geht es auch schon mit dem ersten Teil des Spieles los.

The Paris chase — Verfolgungsjagd in Paris

In diesem Teil müssen Sie mit Ihrem Auto den Landeplatz eines Fallschirmes erreichen. An dem Fallschirm hängt May Day, die im Eiffelturm-Restaurant einen befreundeten Agenten umgebracht hat. Nach dem Mord ist sie dann mit dem Fallschirm vom Eiffelturn gesprungen. Wechselnde Winde machen es Ihnen nicht einfach, den Landepunkt rechtzeitig zu erreichen.

Achtung EUREKA-Fans!

Noch ist Eureka nicht gelöst. Aber die Abenteuerfans kommen in diesen fünf Spielen immer weiter.

Wer versucht hat, den modernen Teil von Eureka zu spielen, dem werden vier Fragen im Wege stehen. Die Antworten auf diese Fragen erhalten Sie in den ersten vier Teilen des Adventures. Für alle, die noch nicht soweit gekommen sind hier die Antworten.
Frage 1: Meep Meep
Frage 2: Nero
Frage 3: Mordred
Frage 4: Die Ambosspolka
Nach der nun folgenden Musik können Sie getrost beginnen.

(R. D. Binggeli jun./rg)

ond — A View to a Kill

Sehen Sie sich den neuen James-Bond-Film nicht nur an. Seien Sie James Bond! In drei unterschiedlichen Situationen müssen Sie beweisen, daß Sie dem Agenten 007 in nichts nachstehen.

Die ersten Vorstellungen und Pläne (siehe Aktuell-Meldung 64'er Ausgabe 7/85) für diesen Teil konnten von den Programmierern dann im Zeitdruck wohl doch nicht ganz umgesetzt werden. Der dreidimensionale Blick aus dem Auto heraus ist etwas mickrig ausgefallen (Bild 1). Alles in allem kann der erste Spielteil nicht ganz überzeugen.

The City Hall escape — Flucht vor dem Feuer

Sie befinden sich im obersten Stockwerk des Rathauses von San Francisco. Max Zorin hat hier in einem der Räume Feuer gelegt. Bond, also Sie, muß versuchen mit seinem Begleiter Stacey aus dem brennenden Gebäude zu entkommen.

Dieser Teil ist ein »Arcade-Adventure« mit recht netter Grafik. Zur Steuerung und Befehlseingabe wird der Joystick benutzt. Gegenstände und Befehle werden in einem »Duck shoot«-Menü angesteuert (Bild 2 oben). Diese Steuerung ist zu Anfang sehr ungewohnt. Aber mit ein wenig Übung ist sie leicht zu beherrschen.

Im Gegensatz zum ersten Spielteil verliert der Spieler nicht so schnell die Lust zum Weiterspielen; man kann sich sogar regelrecht in diese Aufgabe verbeißen.



Titel	A View to a Kill (5 7 9 11 13 15)
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	3 Spiele in einem
Hersteller	Domark
Preis	49 Mark
Bezugsquelle	Rushware Micro Händler

The Mine — Silicon Valley

Dieser Teil spielt in einer Silbermine unter dem Silicon Valley. Hier müssen Sie eine Bombe finden und entschärfen, die Max Zorin versteckt hat. Nur so können Sie das Silicon Valley retten.

Auch diesen Teil kann man mit dem Überbegriff Arcade-Adventure bezeichnen. Von der Machart und Grafik her unterscheidet er sich aber erheblich vom zweiten Teil (Bild 3).

The secret Sequence — Enttäuschung auf der ganzen Linie

Insgesamt bereitet das Spielen der drei Teile viel Spaß. Wer jedoch alle Teile nur durchspielt, um die geheime Sequenz zu sehen, dem wollen wir eine Enttäuschung ersparen. Es lohnt sich wirklich nicht. Daher verraten wir Ihnen an dieser Stelle das Code-Wort. Geben Sie folgende Buchstaben ein und urteilen Sie selbst: ILVCT.

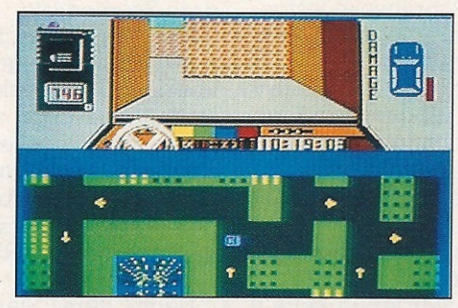


Bild 1. The Paris chase

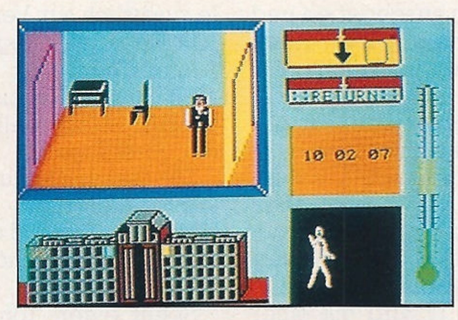


Bild 2. The City Hall escape

Musik von Duran Duran

Zwei Musikstücke sind in dieses Spiel eingebaut. In der Einführung und dem zweiten Spielteil ist dies die bekannte James-Bond-Titelmelodie. Im ersten und dritten Spielteil ist es die spezielle Titelmelodie dieses Filmes von Duran Duran.

Ghostbuster-verwöhnte Ohren werden sicherlich mit der Umsetzung beider Melodien nicht ganz zufrieden sein. Kennt man die klanglichen Fähigkeiten des C 64, fragt man sich automatisch: Warum wurde hieraus nicht mehr gemacht. Trotzdem liegt der Sound des Spieles nicht unter dem Durchschnitt. Man kann zuhören, ohne daß man genervt den Ton abschaltet.

Alles in allem

A View to a Kill ist durch seine drei Teile ein Spiel, das sicherlich vielen Spaß machen wird. Durch die Ankündigungen vor der entgültigen Veröffentlichung waren die Erwartungen zu hoch geschraubt. Hätten die Hersteller »etwas tiefgestapelt« und dieses Spiel nicht so in den Himmel gehoben, wäre dieses wohl besser gewesen.

Zum Schluß noch ein Hinweis für Datasette-Besitzer. Beim Laden von Datasette treten recht häufig Fehler auf. Dies macht sich im Farbwechsel der Turbo-Load-Routine bemerkbar. Hört der ständige Farbwechsel auf, liegt ein Fehler vor. Probieren Sie dann erneut zu laden. Eine weitere Kopie aller drei Teile ist auf der Rückseite der Kassette abgespeichert.

(rg)

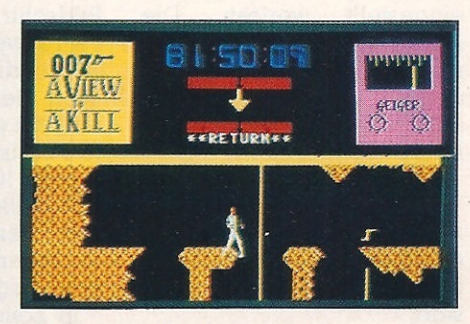


Bild 3. The Silicon Valley Mine

Auflösung: Wettbewerb Bildschirmseite Spitzenprogramme auf einem Bildschirm

Drei Top-Programme aus verschiedenen Themenbereichen stellen wir Ihnen vor.

Eine Flut von Programmeinsendungen erreichte uns auf diesen Wettbewerb. Auch nach intensiver Vorabauswahl kamen noch über 50 Programme in die engere Wahl. Nach weiteren Tests und Diskussionen in der Redaktion blieben schließlich drei Programme übrig. Es handelt sich hierbei um ein Spiel, ein Mathematikprogramm und eine Basic-Erweiterung. Dies veranlaßte uns, ein salomonisches Urteil zu fällen. Wir erhöhten den Preis um 50 Prozent und teilten den Gewinn auf. Jeder der drei Programmautoren erhält 500 Mark. Da die geistige Arbeit, die oft in einem »Bildschirmseitenprogramm« steckt, nicht unerheblich ist, werden wir auch in den nächsten Ausgaben noch einige BS-Seiten veröffentlichen.

Spiel: Block Busters

Block Busters ist eine »Break out«-Version. Sie müssen mit einem Ball möglichst viel von einer Mauer zerstören. Der Ball, der sich auf der Grundlinie hin und her bewegt, kann über den Feuerknopf nach oben geschleudert werden. Jedes zerstörte Mauerelement bringt Punkte. Zehn Bälle stehen Ihnen zur Verfügung. Erreichen Sie 150 Punkte oder ein Vielfaches hiervon, wird eine neue Mauer aufgebaut und Sie erhalten drei Extra-Bälle.

Die Grundanzahl der Bälle kann durch die Veränderung der 10 in Zeile 22, die Anzahl der Extrabälle durch Änderung der 3 in Zeile 14 eingestellt werden. Die Punktanzahl, mit der eine neue Mauer aufgebaut werden soll, kann in Zeile 14 verändert werden, indem man beide 150 verändert.

Noch ein Tip für Anfänger: Wenn Sie die linke oder rechte untere Ecke der Mauer treffen, wird gleich ein großer Teil der Mauer zerstört.

(Martin Greil/rg)

X-Gleichung

Dieses Programm berechnet aus einer X-Gleichung den Wert für X. Durch die Begrenzung auf eine Bildschirmseite mußten leider einige Überprüfungen wegfallen. So muß der X-Wert eindeutig sein, das heißt wenn X +3 und zugleich —3 ist, errechnet das Programm nur die —3. Wenn X unendlich viele Werte annehmen kann, erhält man nur die Lösung X = —1000000. Es dürfen auch keine undefinierten Werte auftreten, sonst erhält man eine Fehlermeldung vom Computer, so zum Beispiel DIVISION BY ZERO bei 1/X und X = 0. Beachtet man dies, kann jede mathematische Funktion, die der C 64 kennt, in der Gleichung verwendet werden.

Programmbeschreibung:

In Zeile 0 bis 4 wird die Gleichung eingegeben und in eine linke (L$) und eine rechte (R$) Seite zerlegt und schließlich in die Zeilen 5 und 6 eingefügt. Nun erfolgt ein Neustart bei der Zeile 5, in der ja der eine Teil der Gleichung definiert ist. In Zeile 7 wird der kleinste Wert für X und in der Variablen I die Schrittweite festgelegt. A$ erhält ein Anführungszeichen und der Bildschirm wird gelöscht. Zeile 8 überprüft, ob die linke Seite der Gleichung mit X = —1E+6 kleiner als die rechte ist. Ist dies nicht der Fall, werden in den Zeilen 12 und 13 die rechte und die linke Seite vertauscht.

Danach werden in Zeile 9 A und B (linke und rechte Seite) definiert. X wird auf dem Bildschirm ausgegeben und I wird, wenn A < B ist, durch 10 dividiert. X nimmt den Wert der Hilfsvariablen V an. Jetzt wird wieder zur Zeile 9 zurückgesprungen.

Zeile 10 prüft, ob A = B ist, das heißt, daß beide Seiten der Gleichung gleich sind. In diesem Fall wird X invertiert ausgegeben.

In Zeile 11 wird mit einer GOSUB-Routine in Zeile 14 abgefragt, ob man durch einen Tastendruck das Programm wieder starten möchte. Dies ist sehr sinnvoll, wenn X beispielsweise gleich ⅓ ist, und es für den Computer keine dezimale Lösung gibt. Wird keine Taste gedrückt, übernimmt V den Wert von X und X wird um I erhöht. Dann erfolgt der Rücksprung zu Zeile 9.

(Christoph Rockenstein/rg)

Musik-Tool

Unser letzter Ein-Seiter ist eine kleine Basic-Erweiterung, die dem C 64 einige neue Sound-Befehle beschert. So etwas ließe sich natürlich nur in Maschinensprache programmieren. Deswegen haben die Autoren zwei trickreiche Eingabesysteme erfunden. Das eine hat den Nachteil, daß es mehr als eine Bildschirmseite benötigt. Dafür ist es aber sehr viel leichter abzutippen als das zweite. Bei der ersten Version wurde das Maschinenprogramm mit den Buchstaben A bis P codiert. Somit kann es jeder innerhalb von wenigen Minuten abtippen. Die zweite Version verwendet den kompletten Commodore-Zeichensatz und ist deswegen sehr schwer einzugeben. Dies zeigt auch das überlange Checksummer-Listing, das beim normalen Listen aber auf eine Bildschirmseite paßt und somit die Wettbewerbsbedingungen erfüllt. Das erste Listing muß übrigens mit RUN 30 gestartet werden.

Nach dem Starten des Musik-Tools stehen Ihnen dann folgende Basic-Befehle zusätzlich zur Verfügung.

SCLR löscht alle Register des Sound-Chips.

ADSR Stimme (1 bis 3), Anschlag, Abschwellen, Halten, Ausklingen

Die Parameter für die Hüllkurve einstellen. Es sind Werte von 0 bis 15 erlaubt.

PLAY Stimme, Tonhöhe, Wellenform

Spielt einen Ton beliebiger Höhe und Wellenform auf einer der drei Stimmen. Die Tonhöhe kann entweder durch einen Wert zwischen 0 und 65535 angegeben werden oder durch die Frequenz in Hertz, wenn man vor die Tonhöhe ein Ausrufezeichen setzt. Es stehen die vier normalen Wellenformen zur Verfügung, deren Werte im Handbuch stehen. Zusätzlich sind die Effekte Synchronisation und Ringmodulation verfügbar. Es muß dann 2 beziehungsweise 4 zur Wellenform addiert werden.

SOFF Stimme

Die Lautstärke der Stimme geht in der Zeit auf Null zurück, die bei ADSR mit Ausklingen bestimmt wurde.

FILT Resonanz + Stimme, Art, Frequenz

Legt die Filterbetriebsart fest. Resonanz + Stimme legt fest, welche Stimmen gefiltert werden sollen und wie hoch die Filterresonanz ist. Dieser Parameter errechnet sich wie folgt:
Wert = Resonanz * 16 + (Stimme —1) hoch 2

Der Wert der Resonanz darf ebenfalls Werte von 0 bis 15 annehmen.

Die Art des Filters wird folgendermaßen bestimmt: 16-Tiefpaß, 32-Bandpaß und 64-Hochpaß. Die Stimme 3 läßt sich stummschalten, wenn man zu diesem Parameter 128 addiert.

Die Filterfrequenz darf Werte von 0 bis 2047 annehmen. Hier ist keine direkte Eingabe in Hertz möglich.

TV Stimme, Tastverhältnis

definiert das Tastverhältnis der Stimme, sofern als Wellenform das Rechteck Verwendung findet. Hier sind Werte von 0 bis 2047 erlaubt.

VOL

legt schließlich die Lautstärke fest, die Werte von 0 bis 15 annehmen darf.

Schließlich wurden noch der POKE- und der PEEK-Befehle angepaßt, so daß sich jetzt auch alle Parameter des SID mit PEEK auslesen lassen.

(Rüdiger Wemski/ Martin Wernecke/bs)

```
0 POKE 53281,0:FOR A=0 TO 57:PRINT TAB(14)
  LEFT$("BLOCK BUSTERS{LIG.RED,WHITE,GREY
  3,GREY 2,GREY 1,BLACK}",B+14)                  <042>
1 B=(B+1)AND 7:NEXT:PRINT TAB(13)"{DOWN,YE
  LLOW}BY MARTIN GREIL":T=54273:POKE T+23,
  15                                             <191>
2 POKE 53280,14:X=1034:Y=15:P=1:V=1:W=1:PO
  KE T+4,51:WAIT 56320,16,240:WAIT 56320,1
  6                                              <135>
4 FOR A=1 TO 36:A$=A$+"V":NEXT:A$="{2SPACE
  }"+A$:PRINT"{WHITE,CLR,YELLOW,4DOWN}":FO
  R A=1 TO 6:PRINT A$:NEXT                       <148>
6 POKE X+V+(Y+W)*40,87:POKE X+Y*40,32            <238>
8 X=X+V:Y=Y+W:POKE T,9*G:POKE T+3,32:POKE
  T+3,33:G=0:IF PEEK(X+V+(Y+W)*40)=32 THEN
   16                                            <127>
10 G=1:P=P+1:IF F=1 THEN W=-W:F=0:GOTO 14        <057>
12 F=1                                           <205>
14 IF P/150=INT(P/150)THEN S=S-3:GOTO 4          <165>
16 IF X+V<1025 OR X+V>1062 THEN V=-V:F=0:G
   =4                                            <011>
18 IF Y+W<1 OR Y+W>23 THEN W=ABS(SGN(Y+W-2
   4)):F=0:G=5:A$="":IF W=0 THEN G=1             <020>
20 IF Y=23 AND PEEK(56320)=111 THEN S=S+1:
   W=-1:G=3:REM"******I(C) BY MARTIN GREIL
   U**                                           <056>
22 IF S<10 THEN 6:"J*********************K       <219>
24 PRINT TAB(5)"{2DOWN}"P"PUNKTE{5SPACE}AL
   TER REKORD"H:WAIT 56320,16,240:IF P>H T
   HEN H=P                                       <219>
26 WAIT 56320,16:S=0:GOTO 2                      <082>
```

Listing »Block Busters«

Variablendefinition

A	: Laufvarariable für Titelgrafik
B	: Für Farbwechsel in der Titelgrafik
X	: X-Koordinate vom Ball +1024 (Startadresse des Bildschirms)
Y	: Y-Koordinate vom Ball
V	: Vektor zur Berechnung der neuen X-Koordinate
W	: Vektor zur Berechnung der neuen Y-Koordinate
T	: Startadresse des SID +1
G	: Zur Berechnung der Tonhöhe
H	: Punkte-Rekord (Hi-score)
P	: Punkte +1 (bei 150 — neue Mauer)
S	: Anzahl der gespielten Bälle (für jede fertiggestellte Mauer —3)
F	: Hilfsvariable zur Ballsteuerung innerhalb der Mauer
A$	: Für Bildschirmaufbau der Mauer

Variablenliste »Block Busters«

Zeile	Bedeutung
0	Titelgrafik
1	Titelgrafik; Initialisierung des SID (Lautstärke)
2	Initialisierung der Variablen
4	Bildschirm löschen und Aufbau der Mauer
6	Löschen und Setzen der Ballposition
8	Berechnung der nächsten Ballposition; Tonerzeugung; Abfrage ob Ball kollidiert.
10 12	Berechnung der neuen Richtung des Balles innerhalb der Mauer
14	Bei erreichen von 150 Punkten (oder ein Vielfaches davon) erhält man 3 Bälle extra; Aufbau einer neuen Mauer
16 18	Richtungsänderung bei Kollision von Ball und Rahmen
20	Hochschnellen des Balles von der Grundlinie bei Betätigung des Feuerknopfes
22	Abfrage, ob die maximale Anzahl der gespielten Bälle schon erreicht ist
24 26	Ausgabe der erreichten Punkte und des Rekords; Behandlung derselben

Programmbeschreibung »Block Busters«

Zeile		
0	POKE53281,0	Farbe Schwarz für Hintergrund
1	T=54273	Tongenerator Frequenz HI
1	POKET+23	Lautstärke 15
2	POKE53280,14	Farbe Hellblau des Rahmens
2	X=1034	Startadresse des Bildschirms +10
2	POKET+4,51	Tongen. Anschlag 3; Abschwellen 3
26;2	WAIT56320,16,240	Warten auf Loslassen des Feuerknopfes (2)
6;8	Zahl 40	ist die Anzahl der Spalten
8	2 POKEs	Ton ausschalten; Ton einschalten
16	Zahl 1025	Bildschirmadr. +1
16	Zahl 1062	Bildschirmadr. —1 + Spaltenzahl
20;18	Zahl 23	Zeilenzahl —1
18	Zahl 24	Zeilenzahl
20	PEEK(56320)=111	Port 2 Feuerknopf gedrückt
24	WAIT56320,16	Warten auf Feuerknopf gedrückt

Hinweise zu »Block Busters« für VC 20-Besitzer

```
0 INPUT"{CLR}GLEICHUNG{HOME,3DOWN}";G$:FOR
   I=1 TO LEN(G$):IF MID$(G$,I,1)="="THEN
  2                                              <118>
1 NEXT:GOTO 0                                    <122>
2 L$=LEFT$(G$,I-1):R$=RIGHT$(G$,LEN(G$)-I)
  :PRINT"{CLR,3DOWN}5DEFFNA(X)=";L$:POKE 1
  98,6                                           <051>
3 PRINT"6DEFFNB(X)=";R$:PRINT"RUN5{HOME}"        <127>
4 FOR I=1 TO 6:POKE 630+I,13:NEXT:END            <201>
5 :                                              <237>
6 :                                              <238>
7 X=-1E+6:I=1E+6:A$=CHR$(34):PRINT"{CLR}";
  :                                              <043>
8 IF FN A(X)<FN B(X)THEN 12                      <203>
9 A=FN A(X):B=FN B(X):PRINT"{HOME}X=";X;"{
  18SPACE}":IF A<B THEN I=I/10:X=V:GOTO 9        <171>
10 POKE 198,0:IF A=B THEN PRINT"{CLR,RVSON
   }X=";X:WAIT 198,1:POKE 198,0:RUN              <143>
11 GOSUB 14:V=X:X=X+I:GOTO 9                     <072>
12 PRINT"{CLR,6DOWN}?"A$"{6UP}5 DEFFNB{8LE
   FT,DOWN}6 DEFFNA{8LEFT,DOWN}RUN5"A$"          <165>
13 FOR I=1 TO 6:POKE 630+I,13:NEXT:POKE 19
   8,6:PRINT"?"A$"{HOME}"A$";{HOME}":LIST
   5-6                                           <189>
14 IF PEEK(198)=0 THEN RETURN                    <215>
15 RUN        20.02.1985 BY CHRI STOP H ROC
   KENSTEIN * BAHNHOFSTR.4 * 8452 HIRSCHAU <228>
```

Listing »X-Gleichung«

```
0 "JKMBAKAMNIIADAMIJADAJKFBAKAMNIKADAMILAD
  AAGAKIACKBCIDALFAJKAAKKIKIBIAEIOJAKBA    <223>
1 "BLKHNNCHAMANKCIOIMNLCHAMJMACANAPIIIJIBF
  GKHFIKHAJCAGOLHIGFKOJKAIKJKHKIEJKNKIE    <158>
2 "JLNJAMIEJLMJAMIEMEDHAAAKBAIONLCHAMJMACA
  NIPIOGOOJNLCHAMANNLICALDAMEEOHKMEGIOK    <148>
3 "BEEEDFCFACGFPEMEACGEJEMEEFACEFGFACAFMEB
  EJFACDFPEGEGEACDFMJACHJACAACMACAAAAFI    <120>
4 "IAMDKFGGKOAMBBBMECBMLGBMKIBMLFBMOKBMONA
  MFLAMACNPOKACOJHLKIAGIGIGACBPOKFKFBIE    <000>
5 "FKEBIEACHPHLFKFBJMENANCBGKEBAOJBALMAMLM
  OBMIGFIEBIGFIFBMECKDLMEGBILACLOHLKIEK    <003>
6 "EBJJMOBMMEICILACNLBMACBABMJJFAENJJBPBMA
  CBABMJJGAENJJCPBMAGACOKAMKAKAKAKAFIFK    <140>
7 "ACOKAMFAFKEKEKAGACOJHLGIEKNKEACMJCAPFAE
  KNIIBENNIEACMAGACOJHLOIHBENOIDACMNKEA    <233>
8 "CMJCPAFICAACOKAMFACANIIBENNIEACMACNPOKA
  CKINKACHPHLFKEBNIFBENNIBACMFKFBJCHANI    <247>
9 "GBENNICACMAGACNLBMIKJLAPBMJCOPJJAPBMJJE
  AENAGACNLBMACNPOKACKINKACHPHLFKEBEKEK    <185>
10 "JJCAENJJOOBMFKFBJJDAENJJPOBMAGACNLBMAC
   PMBMEKEKFKEBJJAAENJJMOBMFKFBJJBAENJJNO  <216>
11 "BMACOKAMEKEKJJAPBMJJEAENAGAKIBJKAAJJAA
   ENJJMOBMIIABHPAGACOJHLKMKIAPIAJKAAIBJG  <013>
12 "HAKMANLPFIEKAGACNPOKJMBCAPGAACKINKMEHP
   HLACDHAAACKINKJKHJAKAMACICKLMEHPHLNEFE  <115>
13 "IEAFIEMEBEDEEEIEBEBEBEDELEJEOELEKELEIE
   KEBELEBENEBEDEJEDELEMENEFEIEAFIEMEOEFE  <136>
30 FOR I=0 TO 13:A=I*82+2054:FOR E=A TO A+
   75 STEP 2:C=PEEK(E)-65+(PEEK(E+1)-65)*1
   6                                       <236>
40 POKE 49152+P,C:P=P+1:NEXT E,I:SYS 49152 <055>
```

Listing »Musik-Tool 1«

```
0 PRINT"{CLR,RVSON,SPACE,RVOFF}@{RVSON}>{R
  VOFF}XD{RVSON}ZJY{RVOFF}@{RVSON}*>{RVOFF
  }XE{RVSON}ZJY{RVOFF}@{RVSON}AHPZ{RVOFF}L
  @{RVSON}*{RVOFF,RVSON}({RVOFF}[{RVSON}←?
  L{RVOFF}GB{RVSON}K{RVOFF}HB{RVSON}({RVOF
  F}T{RVSON}←?L{RVOFF}IB{RVSON}K{RVOFF}";      <225>
1 PRINT"J{RVSON}←{RVOFF}G{RVSON}!{RVOFF,
  SPACE}7{RVSON}/{RVOFF}D{RVSON}(B)'{RVOFF
  }WG{RVSON}C]←{RVOFF}@{RVSON}0{RVOFF}O{RV
  SON}={RVOFF}E{RVSON}?O{RVOFF}){RVSON}MG<
  {RVOFF}E{RVSON}?H{RVOFF}←{RVSON}OPGW{RVO
  FF}W@O{RVSON}D{RVOFF}O{RVSON}O";             <011>
2 PRINT"{RVOFF}A{RVSON}G{RVOFF}GM{RVSON}$]
  {RVOFF}I{RVSON}'(&{RVOFF}G{RVSON}(,{RVOF
  F}G{RVSON}8£?{RVOFF}G{RVSON}8[?{RVOFF}GK
  R{RVSON}B←{RVOFF}@{RVSON}M<{RVOFF}E{RVSO
  N}?H{RVOFF}←{RVSON}OYMG]<{RVOFF}E{RVSON}
  ?O<{RVOFF}'{RVSON}/";                        <126>
3 PRINT"{RVOFF}BK{RVSON}T&{RVOFF}K{RVSON}E
  -{RVOFF}*CRQ←UNK←EHKS←SU←OK*X←RNEE←R{RVS
  ON}[{RVOFF}←{RVSON}V{RVOFF}←{RVSON}BA{RV
  OFF}←{RVSON}BBD{RVOFF}G;YG{RVSON}£?{RVOF
  F}P{RVSON}*";                                <225>
4 PRINT"{RVOFF}#{RVSON}*{RVOFF}N{RVSON}*I*
  {RVOFF}Z{RVSON}*-*-?4?{RVOFF}←{RVSON}C-{
  RVOFF}←{RVSON}]6I{RVOFFMM←{RVSON}↑-${R
  VOFF}TG{RVSON}${RVOFF}SG←{RVSON}T6${RVOF
  F}T{RVSON}HSO{RVOFF}Q{RVSON}%{RVOFF}S{RV
  SON{RVOFF}X";                               <155>
5 PRINT"{RVSON}/{RVOFF}K{RVSON};Q*{RVOFF}M
  {RVSON}D{RVOFF}SM{RVSON}D{RVOFF}TK{RVSON
  }!2{RVOFF}KU{RVSON}7{RVOFF}←{RVSON}N6I#{
  RVOFF}S{RVSON}XQ*{RVOFF}K'{RVSON}7{RVOFF
  }←{RVSON}<*{RVOFF}←@{RVSON}*X{RVOFF}D{RV
  SON}SX↑*{RVOFF}←@{RVSON}*";                  <205>
6 PRINT"X{RVOFF}E{RVSON}SXE*{RVOFF←{RVSO
  N}-?{RVOFF}IIII{RVSON}D${RVOFF}←{RVSON}-
  ?{RVOFF}D{RVSON}$##{RVOFF←{RVSON}]6E#,
  {RVOFF}C{RVSON}A{RVOFF}({RVSON}P{RVOFF}D
  {RVSON}#L{RVOFF}W{RVSON}SL{RVOFF}C{RVSON
  }A{RVOFF←{RVSON}]";                          <133>
7 PRINT"6M{RVOFF}V{RVSON}SM{RVOFF}B{RVSON}
  A,{RVOFF}C{RVSON}A{RVOFF}(N{RVSON}D{RVOF
  F}A←{RVSON}-?{RVOFF}DA{RVSON}L{RVOFF}W{R
  VSON}SL{RVOFF}C{RVSON}A{RVOFF}←{RVSON}C-
  {RVOFF}←{RVSON}I,{RVOFF}←{RVSON}T6${RVOF
  F}S{RVSON}L{RVOFF}T{RVSON}SL{RVOFF}@{RVS
  ON}";                                        <014>
8 PRINT"A${RVOFF}T(F{RVSON}L{RVOFF}U{RVSON
  }SL{RVOFF}A{RVSON}A{RVOFF←{RVSON}<*'8P
  *{RVOFF}({RVSON}XXP*X{RVOFF}C{RVSON}S{RV
  OFF←{RVSON}<*{RVOFF}←{RVSON}C-{RVOFF}←
  {RVSON}I,{RVOFF}←{RVSON}T6${RVOFF}S{RVSO
  N}##X{RVOFF}A";                              <235>
9 PRINT"{RVSON}SXZ*${RVOFF}T{RVSON}X{RVOFF
  }B{RVSON}SXS*{RVOFF←{RVSON}<*{RVOFF}←{
  RVSON}N*##${RVOFF}S{RVSON}XBSXQ*${RVOFF}
  T{RVSON}X{RVOFF}@{RVSON}SXT*{RVOFF}←{RVS
  ON}-?##XP*X{RVOFF}C{RVSON}S{RVOFF";         <249>
10 PRINT"{RVSON}←{RVOFF}W{RVSON}(BXBSXQ*G{
   RVOFF}O{RVSON}T{RVOFF←{RVSON}]6IIP{RV
   OFF}G{RVSON}(B{RVOFF}WFF{RVSON}IOCD#{RV
   OFF←{RVSON}C-H{RVOFF,SPACE,RVSON}P{RV
   OFF}E←{RVSON}I,{RVOFF}K{RVSON}T6{RVOFF}
   ←R{RVSON}";                                 <149>
11 PRINT"B{RVOFF}←{RVSON}I,(V←?{RVOFF}←'{R
   VSON}9{RVOFF}K{RVSON}T6":SYS 4↑5           <155>
```

Listing »Musik-Tool 2«

Interessieren Sie Spiele? Wir bieten über 10000 Mark für Ihre Antwort

Das 64'er-Magazin ist eine leserbezogene Zeitschrift. Das heißt, wir möchten mehr über Sie und Ihre Wünsche erfahren, um diese in die Tat umzusetzen. Dafür haben wir auch diesmal Preise von über 10000 Mark ausgesetzt.

Um an diesem Wettbewerb teilnehmen zu können, müssen Sie nur den hier abgedruckten Fragebogen ausfüllen und an uns zurücksenden. So nehmen Sie nicht nur an der Verlosung der Preise teil, Sie beteiligen sich auch aktiv an der Gestaltung des 64'er-Magazins. Aus der Umfrage können wir dann unter anderem ersehen, ob Ihnen unser neuer Spieleteil gefällt, beziehungsweise ob und was Sie geändert haben möchten. Außerdem wollen wir eine aktuelle Leser-Hitliste der Spiele erstellen.

Als Preise verlosen wir Software im Wert von über 10000 Mark. Die Palette der Programme reicht von Printshop über Paint Magic bis zu Spielen wie Ghost Busters oder Kaiser.

Der Einsendeschluß ist der 15.09.1985. Fragebögen, die nach diesem Termin die Redaktion erreichen, können bei der Verlosung leider nicht mehr berücksichtigt werden.

Der Rechtsweg ist ausgeschlossen!

Die Preise wurden von folgenden Firmen zur Verfügung gestellt:
Ariolasoft
Funtastic
Imagine
Joysoft
Kingsoft
Markt & Technik
Mastertronic
Micro Händler
Rushware

Fragebogen Spiele

1. Welchen Computer besitzen Sie? ☐ C 64 ☐ VC 20 ☐ C 16, C 116 ☐ Plus 4 ☐ CBM 30XX, 40XX, 80XX ☐ ______

2. Besitzen Sie? ☐ Datasette ☐ Floppy

3. Welchen Joystick verwenden Sie? ☐ Quickshot ☐ Comp. Pro Microsw. ☐ Competition Pro ☐ Commodore ☐ ______

4. Wieviel Zeit verbringen Sie am Computer (in Stunden/Tag)? ☐ 1-2 ☐ 2-3 ☐ 3-4 ☐ 4-5 ☐ 5-6 ☐ 6-7 ☐ 7-8 ☐ über 8

5. Wieviel Zeit (in Stunden) verwenden Sie davon für:

☐ Programmieren	☐ unter 1	☐ 1-2	☐ 2-3	☐ 3-4	☐ 4-5	☐ 5-6	☐ 6-7	☐ 7-8	☐ über 8
☐ Spielen	☐ unter 1	☐ 1-2	☐ 2-3	☐ 3-4	☐ 4-5	☐ 5-6	☐ 6-7	☐ 7-8	☐ über 8
☐ Textverarbeitung	☐ unter 1	☐ 1-2	☐ 2-3	☐ 3-4	☐ 4-5	☐ 5-6	☐ 6-7	☐ 7-8	☐ über 8
☐ Datenverwaltung	☐ unter 1	☐ 1-2	☐ 2-3	☐ 3-4	☐ 4-5	☐ 5-6	☐ 6-7	☐ 7-8	☐ über 8
☐ Anwendungen	☐ unter 1	☐ 1-2	☐ 2-3	☐ 3-4	☐ 4-5	☐ 5-6	☐ 6-7	☐ 7-8	☐ über 8
☐ Hardware basteln	☐ unter 1	☐ 1-2	☐ 2-3	☐ 3-4	☐ 4-5	☐ 5-6	☐ 6-7	☐ 7-8	☐ über 8
☐ ______	☐ unter 1	☐ 1-2	☐ 2-3	☐ 3-4	☐ 4-5	☐ 5-6	☐ 6-7	☐ 7-8	☐ über 8

6. Welchen Spieletyp spielen Sie am liebsten?

☐ Text-Adventure	☐ Grafik-Adventure
☐ Fantasy	☐ Rollenspiel
☐ Flugsimulatoren	☐ Simulatoren
☐ Strategie-Taktikspiel	☐ Sportspiel
☐ Schießspiel	☐ Denkspiel
☐ Geschicklichkeitsspiel	☐ Spielgeneratoren
☐ Construction-Sets	☐ Glücksspiele
☐ ______	

7. Wieviele professionelle Spiele besitzen Sie? ______

8. Wieviel Geld würden Sie maximal für ein Spiel, das Sie haben möchten, ausgeben?

☐ Kassette	☐ 10-20	☐ 20-30	☐ 30-40	☐ 40-60	☐ 60-70	☐ 70-80	☐ 80-100	☐ 100-120	☐ 120-140	☐ 140-160	☐ über 160 Mark
☐ Diskette	☐ 15-25	☐ 25-35	☐ 35-45	☐ 45-65	☐ 65-85	☐ 85-105	☐ 105-125	☐ 125-145	☐ 145-165	☐ 165-185	☐ 185-210 ☐ über 210

9. Welche Spiele spielen Sie am liebsten? Bitte Titel und Hersteller angeben.

Titel	Hersteller
1. ______	______
2. ______	______
3. ______	______

10. Spielen Sie hauptsächlich: ☐ allein ☐ mit Freunden ☐ im Familienkreis ☐ Kollegen ☐ Spielgemeinschaft

11. Wo informieren Sie sich über Spiele?
☐ Fachhandel ☐ Kaufhaus ☐ Spielhallen ☐ Messen
☐ Zeitschriften, welche? ______
Welche Rubrik? ☐ Test ☐ Aktuelles ☐ Besprechung ☐ Leserzuschriften
☐ Werbung in: ☐ Fernsehen ☐ Radio ☐ Computer-zeitung ☐ Tageszeitung ☐ Werbe-beilagen ☐ Plakat-werbung ☐ Wurf-sendungen
☐ bei Freunden

12. Welche Kriterien sind für Sie bei einem Spiel wichtig?
______

13. Wie beurteilen Sie den neuen Spieleteil des 64'er-Magazins?
☐ sehr gut ☐ gut ☐ ausreichend ☐ könnte besser sein ☐ mangelhaft
☐ sollte ausgeweitet werden ☐ ist in Ordnung ☐ ist zu umfangreich

14. Was könnte verbessert werden?
______

Name: ______ Straße: ______

Ort: ______ Alter: ______

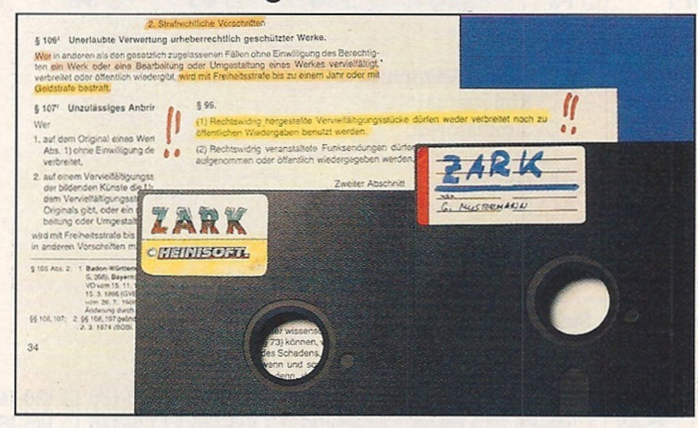
Beruf: ______ Schulbildung: ______



FRAGEBOGEN: SPIELE (Bitte ausfüllen und einsenden)

Änderung des Urheberrechts

Ins Urheberrechtsgesetz sind nun auch Programme für die Datenübertragung aufgenommen worden. Allerdings hat der Gesetzgeber keine Hinweise gegeben, wie die Verletzung des neuen Rechts festzustellen ist, und wie dieses Recht durchgesetzt werden kann.



Aus diesem Grunde ist es erforderlich, sich darüber zu informieren, wie die damit befaßten Gerichte diese Problematik behandelt haben. Denn tatsächlich ist in der Vergangenheit es nahezu unstreitig gewesen, daß die Software dem Schutz des Urheberrechts unterliegt. Äußerst umstritten ist hingegen das Problem, in welchem Umfang der Schutz der Computerprogramme besteht. Unterschieden wird nach der Rechtsprechung, ob sich der Schutz lediglich auf die äußere Form beschränkt, oder ob es allein auf die Idee und den logischen Ablauf ankommt (OLG Karlsruhe in GRUR 1983, 300/306; LG München in GRUR 1983, 175/176). Darüber hinaus gibt es auch Entscheidungen, die sowohl die Form als auch den Inhalt für schutzwürdig halten und letztlich darauf abstellen, was im konkreten Einzelfall als schutzbegründend angesehen wird (OLG Hamburg in GRUR 1983, 436).

Aus den bisher vorliegenden Urteilen kann man auch die Schwierigkeit der Feststellung des Inhaltsschutzes entnehmen. Darüber hinaus besteht insoweit auch noch das Problem, den Inhaltsschutz in einer angemessenen Zeit gerichtlich durchzusetzen. Hierbei ist das Gericht allein auf die Sachverständigen angewiesen, die aber eine ganz andere Denkweise haben, als die Juristen (Brandi-Dohrn in GRUR 1985, 179, 183).

Es liegt auf der Hand, daß die Erstellung von Sachverständigengutachten sehr zeitaufwendig ist. Demzufolge ist häufig die wirtschaftliche Entwicklung über das streitige Programm hinweggegangen, bevor der Rechtsschutz eingegriffen hat (Brandi-Dohrn a.a.O.).

Aber auch der Formschutz ist begrenzt. Zum Beispiel dann, wenn der Urheber sein altes Programm so umarbeitet, daß die alte Form hinter der neuen verblaßt. Der Angestellte, der ausscheidet, vorher aber eine Kopie des Quellencodes an sich nimmt und nun seine eigene alte Handschrift ändert, in dem er Bearbeitungsschritte austauscht, aufspaltet, Unter-Unterprogramme direkt an das Hauptprogramm anbindet und umgekehrt, mag gegen § 17 WWG verstoßen, kann aber urheberrechtlich durchaus frei bleiben (Brandi-Dohrn a.a.O.).

Es darf auch nicht übersehen werden, daß der Erwerber eines gemäß Urheberrechts verletzend nachgebildeten Computerprogramms, auch wenn er bei dem Erwerb gutgläubig war, seinerseits eine Urheberrechtsverletzung begeht, wenn er das Programm weiter vervielfältigt oder weiter verarbeitet. Wenn es sich hierbei um eine Vervielfältigung nach § 16 UrhG handelt, kann der Urheber oder der entsprechende Rechtsinhaber die weitere Benutzung untersagen. Das ist allerdings in dem Bereich, in dem Raubkopien verwendet werden, kaum möglich und sinnvoll, da es dort zuviele Benutzer gibt (Brandi-Dohrn a.a.O.).

Rechtlich kann das dazu führen, daß jeder Interessent damit rechnen muß, daß er nach dem Ankauf eines Computerprogrammes wegen Unterlassung oder Schadenersatzes in Anspruch genommen wird. Das kann den Markt nachhaltig beeinträchtigen. Der BGH (BGHZ 17, 266/269) hat hierzu entschieden, daß es sich bei Vervielfältigungen im Sinne des § 16 UrhG um die körperliche Festlegung eines Werkes handelt. Aus diesem Grunde erstreckt sich daher die Reichweite des Urheberrechts nur auf Vervielfältigungsstücke in gesonderter körperlicher Form von Disketten, Platten oder Bändern und deren Verbreitung, nicht aber auf die Nutzung durch bloßen Programmablauf (Brandi-Dohrn a.a.O.).

Letztlich ist es so, daß demzufolge nur in dem vorgenannten Rahmen ein Urheberrechtsschutz stattfindet.

Äußerst problematisch bleibt auch nach der Novellierung des Urheberrechtsgesetzes die Durchsetzung des Rechts des Verletzten beim Gericht. So muß der Verletzte zunächst einmal an den Verletzer herankommen, was nicht immer ganz einfach möglich ist, da der Verletzer nicht unbedingt gleich der Abnehmer ist. Darüber hinaus muß dann vor dem Gericht im einzelnen dargelegt werden, ob die Behauptungen des Verletzten zutreffend sind oder nicht, was bedeutet, daß der angebliche Verletzer, um sich zu schützen, sein eigenes Programm darlegen muß, was selbstredend aber außerordentlich nachteilig für ihn sein kann.

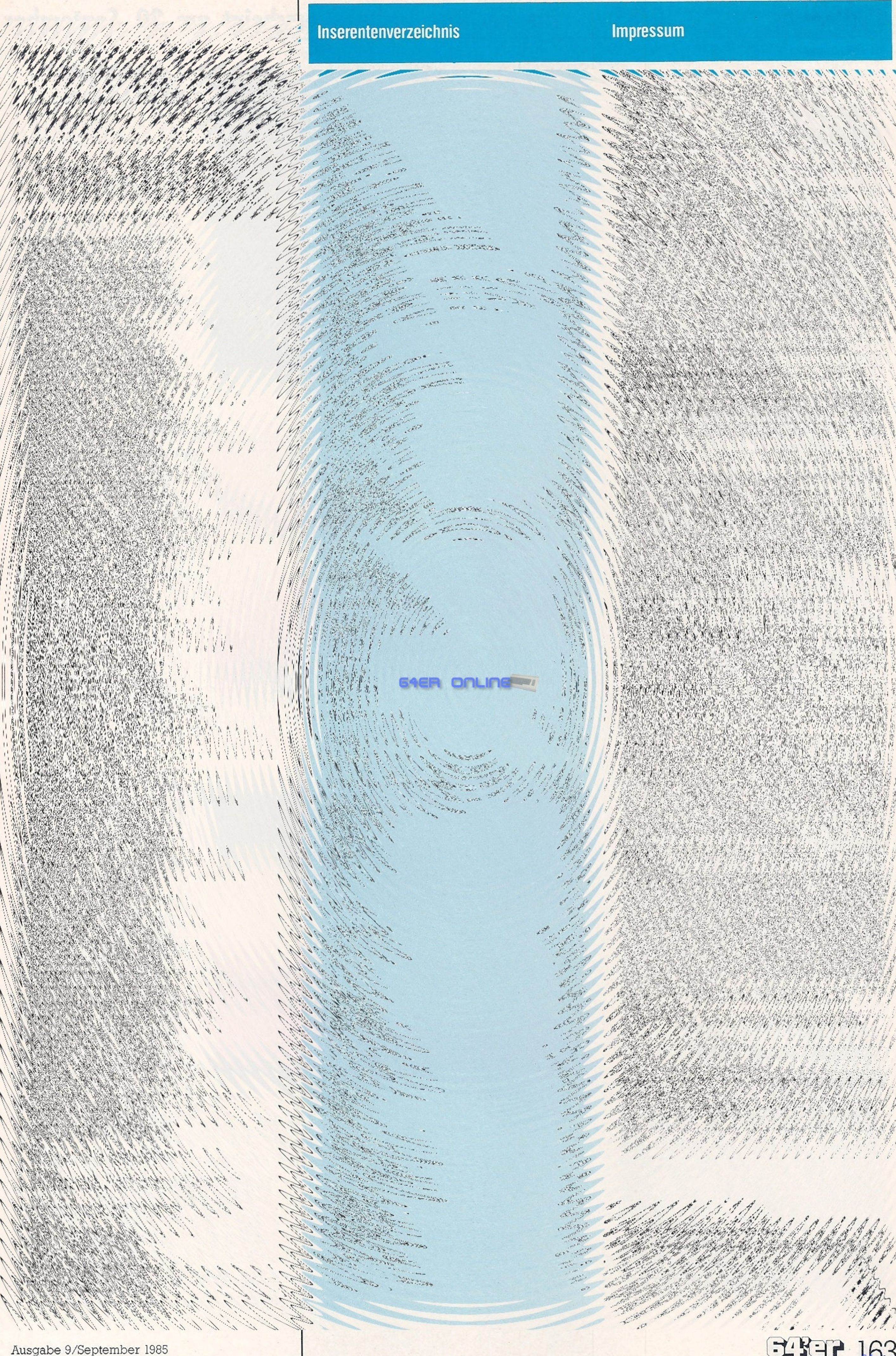
Wenn es dann über den einstweiligen Rechtsschutz hinaus zu einer Hauptsachenklage kommt, ist es meistens schon — zeitlich bedingt — so weit, daß der angeblich Verletzte entweder kein Interesse mehr an der Rechtsverfolgung hat, oder der Verletzer »neue Releases« zu dem Programm herausbringt. Das, was das Gericht ihm schließlich nach der alten Klage zur Zeit der letzten mündlichen Verhandlung verbietet, ist dann oft nicht mehr das, was der Beklagte im x-ten Release zwischenzeitlich auf den Markt bringt (Brandi-Dohrn a.a.O.).

Nach alledem gibt es bisher keinen absolut wirksamen Rechtsschutz. Dieser dürfte wohl nur durch umfangreiche Änderungen gesetzlicher Art, auch im prozessualen Bereich, möglich werden.

Nicht vergessen darf man, daß gleichzeitig auch die Strafbestimmungen des Urheberrechtsgesetzes neu gefaßt worden sind. So ist nunmehr ein § 108a eingeführt, wonach die gewerbsmäßige Vervielfältigung oder Verbreitung urheberrechtlich geschützter Werke ohne Zustimmung des Rechtsinhabers mit Freiheitsstrafe bis zu 5 Jahren oder Geldstrafe geahndet wird. Hierbei ist sogar der Versuch strafbar.

Darüber hinaus ist § 109 geändert worden. Nach der alten Vorschrift wurde Urheberrechtsverletzung nur auf Antrag verfolgt. Dies ist jetzt dahingehend ausgedehnt worden, daß bei besonderem öffentlichen Interesse die Ermittlungsbehörden von Amts wegen einschreiten müssen. Aber auch hier wird es problematisch bleiben, da es bislang kaum Staatsanwälte gibt, die die Sachkunde haben, festzustellen, ob eine Urheberrechtsverletzung im Softwarebereich von besonderem öffentlichen Interesse ist.

(RA Küttner/aa)



Vokabeltrainer

Corsus Latinus ist eine wirksame und interessante Hilfe beim Pauken von Vokabeln. Durch eine klare Benutzerführung und sehr gute Realisierung lernt man schon bei der Eingabe der Daten. Dabei können nicht nur eine einfache Übersetzung, sondern auch mehrere Bedeutungen abgefragt werden. Corsus Latinus ist eine typische Anwendung des Monats.

Die Druckerparade

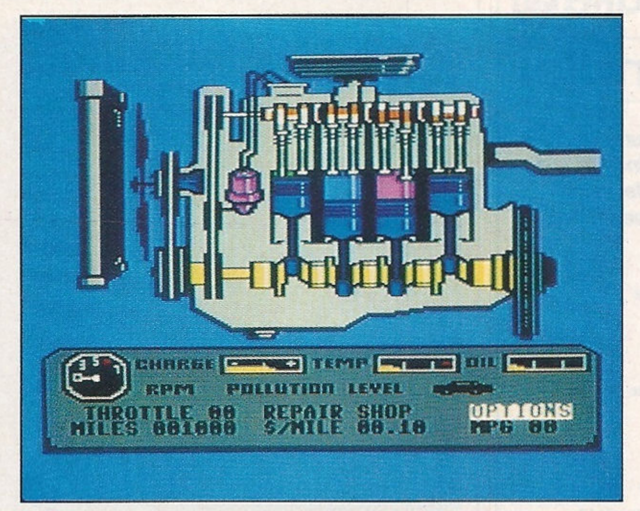
Wieviel Geld muß man für einen guten Matrixdrucker ausgeben? Wir stellen Ihnen drei neue Drucker verschiedener Preiskategorien für Heim und Büro vor. Der Epson GX 80 überrascht mit einem völlig neuen Schnittstellen-Konzept. Der Melchers DX 80X zeigt, daß auch in einem kompakten Gehäuse große Leistung stecken kann. Ein bisher unbekannter Drucker ist der RFI DP-165 — zu unrecht wie wir meinen.

Hypra-Text

Wieder einmal können wir Ihnen ein Listing des Monats par excellence bieten, gekennzeichnet durch den Vorsatz Hypra. Hypra-Text ist ein Textverarbeitungsprogramm, das mit anderen professionellen Textprogrammen durchaus mithalten kann. Einfache Bedienung und Schnelligkeit sind hervorragend realisiert worden.

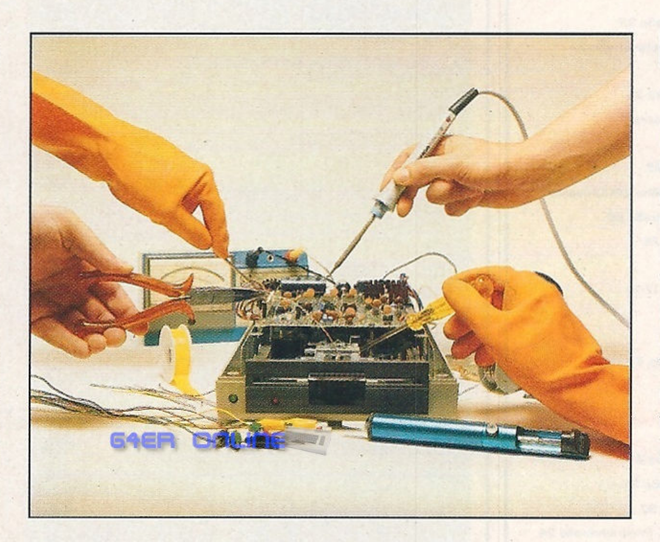
Modulplatinen zum Selberbauen

Fertigen Sie sich Ihre eigenen Steckmodule — wir zeigen Ihnen wie. Mit unserer Bauanleitung können Sie sich mit wenig Aufwand eine 1 x 8 KByte- und 2 x 8 KByte-Modulplatine für den Expansion-Port selbst herstellen. In diesem Artikel erfahren Sie ebenfalls, wie Ihre Programme auf EPROM gebrannt werden.



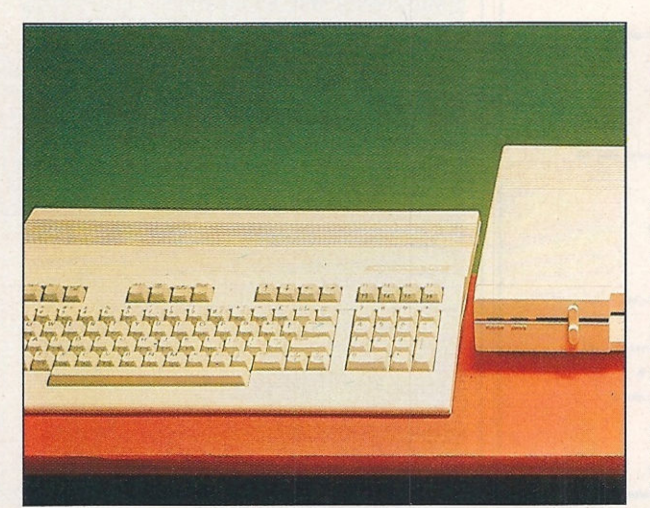
Lernsoftware

Vokabelprogramme gibt es viele. Doch damit ist der Bereich Lernsoftware noch nicht abgedeckt. Hinzu kommen Simulationsprogramme, Computerkurse und Feriencamps. Für alle, die sich für dieses Gebiet der Software interessieren, bringen wir in Ausgabe 10 eine Marktübersicht und Hintergrundinformationen.



Datenspeicherung

Datasette und 1541-Floppy sind nicht mehr die einzigen Massenspeicher für den VC 20, C 64 und C 128. Sie erfahren, unter anderem, wie sich die SFD 1001 im Dauertest (Mailboxbetrieb) bewährt hat, welche Möglichkeiten das 8250-Laufwerk bietet und welche neuen Entwicklungen es zur Beschleunigung der 1541 gibt.



Außerdem

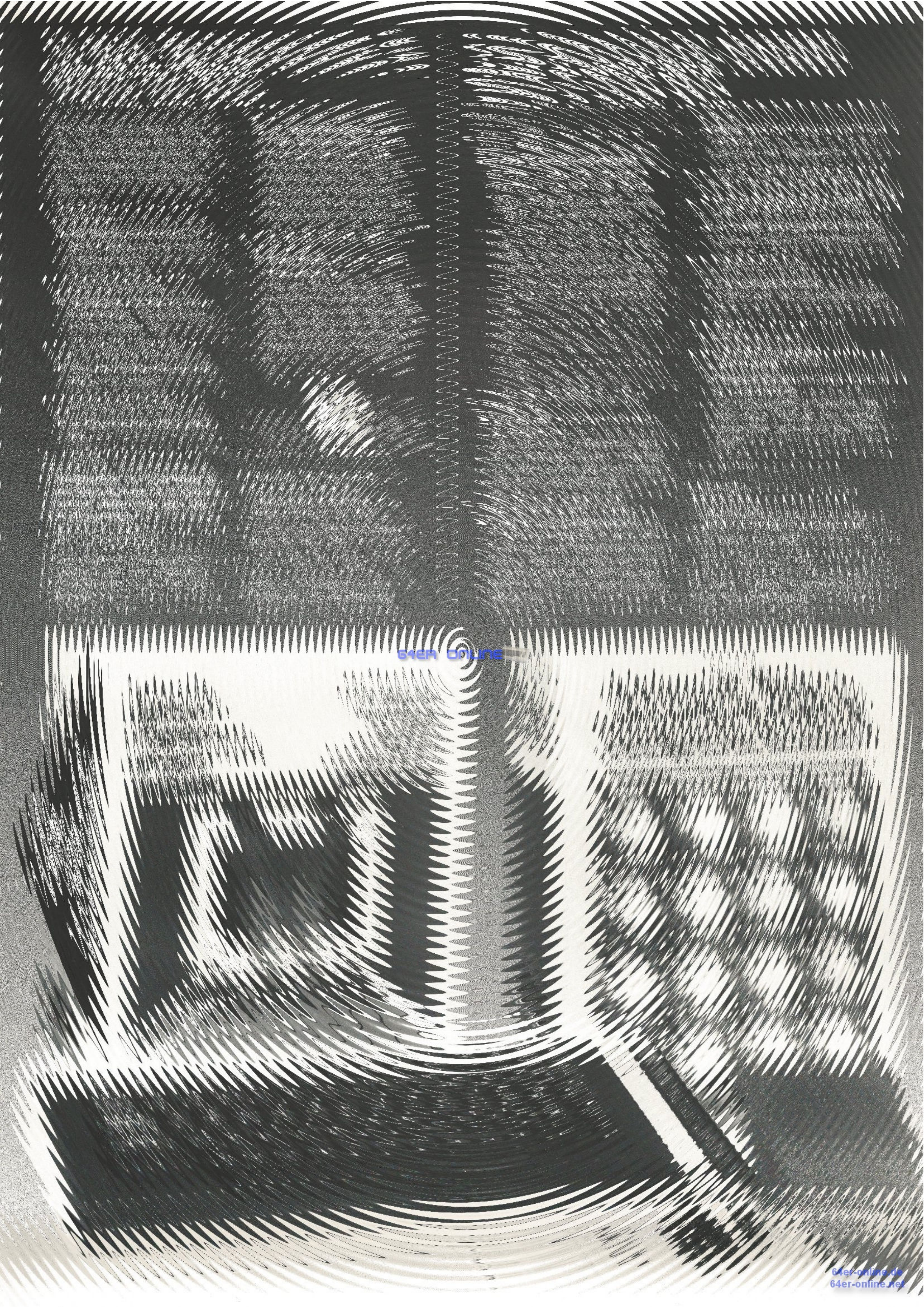
— 64'er Extra. Diesmal: Der VIC-Chip im Detail
— Comal 80, endlich das Comal-Modul für den C 64
— Erweiterungen und Verbesserungen zu Hypra-Ass, Hypra-Load, SMON und XREF
— Directory-Manipulationen II
— und wieder viele Tips und Tricks für den C 64.

Der C 128 im 64'er-Magazin

Nach monatelangen Vorbereitungen kommt er diese Tage endlich auf den Markt: Der neue Commodore 128. Als besonderen Service für alle, die sich für den C 128 interessieren (und natürlich für alle, die ihn bereits haben), finden Sie ab nächsten Monat regelmäßig ein »Magazin im Magazin«, randvoll mit News, Informationen, Tips & Tricks zum C 128. Die Schwerpunkte dieses C 128-Sonderteils im Oktober sind:

— Kompatibilität zum C 64-System
— Einführung in das Basic 7.0
— Fragen und Antworten zum C 128
— Software für den C 128-Modus
— Was gibt es an CP/M-Software?
— Kurs: Das CP/M-Betriebssystem

... und natürlich Tips und Tricks zum C 128.




64ER ONLINE